OGC城市地理标记语言（CityGML）编码标准

OGC China

2021年11月4日

国际开放地理信息协会

提交日期: <2008-05-19>

批准日期: <2012-03-09>

出版日期: <2012-04-04>

本OGC®文件的外部标识符: http://www.opengis.net/spec/citygml/2.0

本OGC®文件的内部参考号: 12-019

类别: OGC®编码标准

编辑: Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, Karl-Heinz Häfele

OGC城市地理标记语言（CityGML）编码标准

文件类型:	OGC®编码标准
文件子类型:	编码
文件阶段:	批准公开发布
文件语言:	中文

OGC China（OGC中国联盟）简介

OGC，英文全称Open Geospatial Consortium，中文可以称国际开放地理信息协会，1994年成立，是一个国际化的自愿协商的组织联盟，旨在通过一系列的敏捷协作研发和开放标准制定，解决地理空间与位置信息可发现、可访问、可互操作、可重用中面临的挑战，提升地理空间信息与位置服务的能力，全球各国地理信息相关人员、组织都可以申请加入。

2017年，经与OGC亚太区负责人沟通，OGC中国联盟成立。经过两三年的发展，从最初联盟发起时的6家单位发展到现在的20多家会员，包括国内代表性的高校，国内主要的地理信息厂商等，旨在促进中国对地理空间软件和数据开放标准的使用和理解，并影响这些标准的定义，同时为空间数据产品和服务的开发人员和用户提供一个协作的平台，共同制定一套符合中国要求的标准。我们希望通过OGC中国联盟，交流经验，无论是高校还是企业，都能在OGC技术工作中发现问题，参与更多的工作组工作，一起在国际标准制定中发挥更大作用。

OGC中国联盟的工作计划：

- 根据中国国家政策，培养对地理空间互操作性最佳实践的认识和实施
- 开发一套中国领导的OGC标准，适应中国的问题、用例和组织特性
- 定期协调中国GIS学会年会互操作会议的组织工作
- 推动更多的中国组织机构包括企业、政府机构、高校加入OGC中国联盟，提升中国国际话语权、影响力
- 支持OGC的全球推广和亚太地区项目，以进一步推进OGC的全球使命

联系方式：

电话：027-6878169
Email：pyue@whu.edu.cn
邮编：400379
地址：湖北省武汉市珞喻路129号武汉大学遥感信息工程学院
官方公众号：国际开放地理信息OGC联盟
网址: https://external.ogc.org/twiki_public/ChinaForum

译者序

当前，全国各地均在积极规划和推进三维城市的建设，从数字城市到智慧城市，再到数字孪生城市，随着信息技术和信息采集技术日新月异，三维城市的内涵和理念也在不断的延伸和丰富。数据是三维城市建设的基石，然而，现阶段三维城市建设中采用的三维模型多以摄影测量技术采集而得的地物表面模型为主，较注重三维可视化效果，而缺乏语义和拓扑关系的表达，难以用于查询、分析或空间数据挖掘，并且采用数据格式缺乏一致性，导致互操作性困难，抑制了三维城市模型的进一步应用与发展。

OGC CityGML标准的提出为虚拟三维城市模型的可重用提供了一个有效的方案。CityGML是一种基于XML的虚拟三维城市模型的存储和交换格式，以模块化的方式定义了三维城市模型中最常见的对象，并兼顾了城市中三维对象的几何、拓扑、语义、外观等方面的属性。CityGML所有对象和主题类均支持多级别细节层次模型，适用于三维城市的多尺度表达，既可表示没有拓扑和语义的单一简单模型，也可以表示具有完整拓扑和细粒度语义的复杂多尺度模型。除定义的建筑物，桥梁、植被、水体、城市家具等三维城市最重要的对象外，CityGML也可通过定义应用领域扩展（ADEs）将其扩展到特定的领域，例如城市气体污染扩散模拟、光辐射潜力分析、城市环境噪声扩散模拟、视线和阴影分析等。CityGML已经在全世界范围内被广泛传播，如德国、法国，美过、英国、荷兰、丹麦、苏黎世、利沃登、日本、伊斯坦布尔等国家或城市，均使用CityGML进行三维城市模型的表达和交换，支撑其三维地理信息标准和三维空间数据基础设施的建设。许多商业和学术工具也通过接口的方式来支持CityGML的使用，例如，SAFE软件公司的转换工具FME、Bentley Systems的BentleyMap、Autodesk的LandXplorer、KIT Karlsruhe的FZKViewer、以及CPA Geo Information的SupportGIS等工具均支持CityGML的读写。因此，通过CityGML标准，可以轻松实现虚拟三维城市模型在不同地理信息系统和用户之间的信息无损交换。

在全国各地都在稳步推进智慧城市或数字孪生城市的今天，CityGML这种通用虚拟三维城市模型存储和和交换格式标准的重要性愈加凸现。鉴于现有CityGML英文标准给国内技术人员带来了一定的语言门槛，也为推动CityGML标准更好服务于国内的智慧城市和数字孪生城市的建设，OGC China翻译了OGC CityGML编码标准2.0版本，向国内GIS及相关行业的科研和工作人员提供自由免费的中文版本。

本标准翻译工作的情况如下：林旭辉（英国伦敦大学校友会）负责第0-3章节的翻译和排版、第10章节的翻译和排版，以及附录A-K的翻译；于大宇（武汉大学）参与了所有格式和内容的审校及所有章节的排版；张勇（郑州众合景轩信息技术有限公司）负责第7-9章节的翻译和排版，以及附录A-K的排版；黄栋（泰瑞数创科技（北京）有限公司）负责第4-6章节的翻译和排版；杨滔、罗维祯和邓成汝协助了林旭辉的翻译和排版工作；乐鹏（武汉大学）负责翻译工作的组织协调、统稿与修订。

虽然译者尽量地以“信、达、雅”为标准还原作者的本意，但仍难免有疏漏之处，如读者发现任何不理想之处，望能不吝赐教，我们将随时予以纠正。

OGC中国联盟
2021年11月4日

本文件引用格式：OGC中国联盟, 2021. OGC城市地理标记语言（CityGML）编码标准中文版. 译者: 林旭辉, 于大宇, 张勇, 黄栋, 乐鹏, 杨滔, 罗维祯, 邓成汝.

许可协议

国际开放地理信息协会（“许可方”）特此授权获得本知识产权及任何相关文件副本的任何人，在不受限制的情况下（下文规定的除外）处理本知识产权，允许获得本知识产权的人具有包括但不限于实施、使用、复制、修改、合并、出版、分发和/或转授本知识产权副本的权利，但必须完整地保留本知识产权上的所有版权声明（译注：此处版权声明指英文原版），且同意本协议的所有条款。

如果您修改了本知识产权，除了上述版权声明外，经修改的知识产权的所有副本必须包括一项声明，以表明该知识产权包括未经授权人批准或采纳的修改。

本许可仅为版权许可，不传达可能在世界任何地方生效的任何专利下的任何权利。

本知识产权“按原样”提供，不作任何明示或暗示的保证，包括但不限于对适销性、特定用途的适用性和不侵犯第三方权利的保证。本声明中包含的一个或多个版权持有人不保证本知识产权中包含的功能将满足您的要求，也不保证本知识产权的运作不会中断或没有错误。对本知识产权的任何使用应完全由用户自己承担风险。在任何情况下，版权持有人或本知识产权的任何贡献者对任何索赔，或任何直接、特殊、间接后果性损害，或因实施、使用、商业化或履行本知识产权而引起的或与之相关的任何涉嫌侵权或任何数据、利润的损失（无论是在合同诉讼、疏忽或任何其他法律理论下）而造成的任何损害概不负责。

本许可在终止之前一直有效，您可以在任何时候通过销毁本知识产权及任何形式的副本来终止它。如果您未能遵守本协议的任何条款或条件，本许可也将终止。除下述规定外，本许可证的终止不要求终止自终止通知之日起生效的任何第三方最终用户对本知识产权的再许可。此外，如果该知识产权或该知识产权的运作侵犯了或在许可方看来可能侵犯了第三方的任何专利、版权、商标或其他权利，您同意许可方可自行决定终止本许可，而无需对您、您的被许可人及任何其同伴进行任何赔偿或承担责任。您同意在许可任何形式终止后，销毁本知识产权以及任何形式的所有副本，无论是由您还是由第三方持有。

除本通知所包含的内容外，未经许可方或任何其他知识产权版权持有人的事先书面授权，许可方或任何其他知识产权版权持有人的姓名不得用于广告或以其他方式促进本知识产权的销售、使用或其他交易。许可方是且在任何时候都唯一可以授权您或任何第三方使用认证标志、商标或其它可表明符合任何许可方标准或规范的特殊名称。本协议受马萨诸塞州法律管辖。特此明确排除《联合国国际货物销售合同公约》对本协定的适用。如果本协议的任何条款被视为不可执行、无效或无效，则应修改该条款以使其有效和可执行，经修改后，整个协议应保持完全效力。许可方的任何决定、作为或不作为均不得被解释为放弃其享有的任何权利或补救措施。

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’ S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’ s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any

LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

目录

i. 摘要

CityGML是一个基于XML的开放数据模型，主要用于存储和交换虚拟3D城市模型。它是地理标记语言（版本号v3.1.1，GML3）的一个具体应用模式, 由国际开放地理信息协会（OGC）和国际标准化组织211技术委员会（ISO TC211）发布的，针对于空间数据交换的可扩展国际标准。CityGML开发的目标是实现对三维城市模型的基本实体、属性和关系的共同定义。该格式对于3D城市模型的低成本可持续维护尤其重要，允许相同数据在不同领域里的复用。

ii. 前言和致谢

上图为CityGML的官方标识。为浏览CityGML最新新闻及资讯，以及关于CityGML领域正在进行的项目和研究领域，请参阅 http://www.citygml.org 和 http://www.citygmlwiki.org 。

CityGML的工作由OGC三维信息管理（3DIM）工作组进行讨论和协调，其最初是作为OGC创新计划第四期网络服务（OWS-4）中CAD/GIS/BIM专题的一部分进行实施和评估。

本标准文件的2.0版由OGC CityGML标准工作组（SWG）编写。未来的讨论和发展将由3DIM工作组领导。

欲了解更多信息，请访问 http://www.opengeospatial.org/projects/groups/3dimwg 。

CityGML也将继续由德国空间数据基础设施（GDI-DE）的3D特别兴趣小组（SIG 3D）成员与OGC中的3DIM工作组和CityGML SWG联合开发。

欲了解更多信息，请访问 http://www.sig3d.org/ 。

英文文件的编写和欧洲讨论会议得到了欧洲空间数据研究组织（EuroSDR，前身为OEEPE）第III委员会项目的支持。

欲了解更多信息，请访问 http://www.eurosdr.net 。

iii. 提交组织

本国际标准是由OGC的CityGML 1.0 SWG成员提交至OGC。其中包括以下组织：

1. Autodesk，Inc.（主要提交方）。
2. Bentley System, Inc （主要提交方）。
3. 柏林技术大学（技术提交方）。
4. 英国军械测量局。
5. 德国波恩大学。
6. 德国哈索·普拉特纳研究所-波茨坦大学IT系统工程研究所。
7. 卡尔斯鲁厄理工学院应用计算机科学研究所。

CityGML最初由SIG 3D开发，2002-2012- http://www.citygml.org 。

iv. 提交组织的联系方式

有关本文档的所有问题可直接向编辑或贡献者(包括开发人员，参见第V条)提出：

姓名	组织	邮箱
Prof. Dr. Thomas H.Kolbe	Institute for Geodesy and Geoinformation Science, Technical University Berlin	thomas.kolbe@tu-berlin.de
Claus Nagel		claus.nagel@tu-berlin.de
Alexandra Lorenz		alexandra.lorenz@tu-berlin.de
Dr. Gerhard Gröger	Institute for Geodesy and Geoinformation, University of Bonn	groeger@ikg.uni-bonn.de
Prof. Dr. Lutz Plümer		pluemer@ikg.uni-bonn.de
Angela Czerwinski		czerwinski@ikg.uni-bonn.de
Haik Lorenz	Autodesk, Inc.	haik.lorenz@autodesk.com

姓名	组织	邮箱
Alain Lapierre	Bentley Systems, Inc.	alain.lapierre@bentley.com
Stefan Apfel		frank.steggink@bentley.com
Paul Scarponcini		paul.scarponcini@bentley.com
Carsten Rönsdorf	Ordnance Survey, Great Britain	Carsten.Roensdorf@ordnancesurvey.co.uk
Prof. Dr. Jürgen Döllner	Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam	juergen.doellner@hpi.uni-potsdam.de
Dr. Joachim Benner	Institute for Applied Computer Science, Karlsruhe Institute of Technology	joachim.benner@kit.edu
Karl-Heinz Häfele		karl-heinz.haefele@kit.edu

v. 开发人员

姓名	组织
Ulrich Gruber	District Administration Recklinghausen, Cadastre Department, Germany
Sandra Schlüter	
Frank Bildstein	Rheinmetall Defence Electronics, Germany
Rüdiger Drees	T-Systems Enterprise Services GmbH, Bonn, Germany
Andreas Kohlhaas	GIStec GmbH (formerly), Germany
Frank Thiemann	Institute for Cartography and Geoinformatics, University of Hannover
Martin Degen	Cadastre Department, City of Dortmund
Heinrich Geerling	Architekturbüro Geerling, Germany
Dr. Frank Knospe	Cadastre and Mapping Department, City of Essen,
Hardo Müller	Snowflake Software Ltd., Great Britain
Martin Rechner	rechner logistic, Germany

姓名	组织
Jörg Haist	Fraunhofer Institute for Computer Graphics (IGD), Darmstadt, Germany
Daniel Holweg	
Prof. Dr. Peter A. Henning	Faculty for Computer Science, University of Applied Sciences, Karlsruhe, Germany
Rolf Wegener	State Cadastre and Mapping Agency of North-Rhine Westphalia, Germany
Stephan Heitmann	
Prof. Dr. Marc-O. Löwner	Institute for Geodesy and Photogrammetry, Technical University of Braunschweig
Dr. Egbert Casper	Zerna Ingenieure, Germany
Christian Dahmen	con terra GmbH, Germany
Nobuhiro Ishimaru	Hitachi, Ltd., Japan
Kishiko Maruyama	
Eiichiro Umino	
Takahiro Hirose	
Linda van den Brink	Geonovum, The Netherlands
Ron Lake	Galdos Systems Inc., Canada
David Burggraf	
Marie-Lise Vautier	Institut géographique national, France
Emmanuel Devys	
	Mark Pendlington

vi. 对OGC抽象规范的更改

该OGC®标准遵循现有的OGC®抽象规范。

vii. 致谢

SIG 3D组织感谢OGC的CityGML SWG和3DIM工作组成员, 以及所有提出修改请求和意见的贡献者。

如下：

```
Tim Case, Scott Simmons, Paul Cote, Clemens Portele, Jeffrey Bell, Chris
Body, Greg Buehler, François Golay, John Herring, Jury Konga, Kai-Uwe
Krause, Gavin Park, Richard Pearsall, George Percivall, Mauro Salvemini,
Alessandro Triglia, David Wesloh, Tim Wilson, Greg Yetman, Jim Farley,
Cliff Behrens, Lukas Herman, Danny Kita, Simon Cox
```

感谢对本标准进行仔细审阅并评论的贡献者：

Ludvig Emgard, Bettina Petzold, Dave Captick, Mark Pendlington, Alain Lapierre, Frank Steggink.

序言

NOTE

请注意，本文件的某些内容涉及到专利权的问题。OGC不承担任何或识别出任意一项专利的责任。到目前为止，还没有人对该情况进行申辩。为避免该标准的发放导致相关知识产权受损，请对该文本保持有疑义的读者提交对本标准的意见，提交相关专利声明或知识产权相关的内容，并提交对应文件。

CityGML 2.0.0版本和CityGML 1.0.0版本（OGC doc No.08-007r1）之间的重大变化：

- 表达隧道和桥梁表示的新主题模块；
- 用于建筑和建筑部件外壳语义分类的附加边界表面（*OuterCeilingSurface*, *OuterFloorSurface*）；
- 建筑物和建筑部件中的底面轮廓和屋顶边界的LOD0表达；
- 表达城市对象相对于周围地形和水面位置的附加属性（*relativeToTerrain*, *relativeToWater*）；
- 测量值和属性集的附加通用属性。
- CityGML代码列表机制的重新设计（枚举属性现在属于 *gml：codetype* 类型，有助于提供枚举其可能属性值的附加代码列表）。

将现有的CityGML 1.0实例迁移到有效的2.0实例，只需要将文档中的CityGML命名空间和模式位置值更改为CityGML 2.0即可。

0. 引言

0.1. 动机

越来越多的城市和公司正在为城市规划、移动通信、灾害管理、三维地籍、旅游、车辆和行人导航、设施管理和环境模拟等不同的应用领域构建虚拟三维城市模型。此外，在欧洲环境噪声导则（END，2002/49/EC）的实施过程中，三维地理信息和三维城市模型发挥了重要作用。

近年来，大多数虚拟三维城市模型被定义为纯图形或几何模型，忽略了语义和拓扑方面的内容。因此，这些模型几乎只能用于可视化目的，而不能用于主题查询、分析任务或空间数据挖掘。由于模型的有限可重用性抑制了三维城市模型的广泛应用，因此必须采用一种更通用的建模方法，以满足不同应用领域的信息需求。

CityGML是一种用于表示三维城市对象的通用语义信息模型。CityGML可在不同应用程序之间共享，对于3D城市模型的低成本、可持续维护尤为重要，这可允许相同数据在不同领域里的复用。可显著应用CityGML的领域包括城市规划、建筑设计、旅游休闲活动、环境模拟、移动通信、灾害管理、国土安全、房地产管理、车辆和行人导航、培训模拟器等。

CityGML是一种基于XML的格式地开放数据模型，用于虚拟三维城市模型的存储和交换。它是作为GML3的一个具体应用模式实现，GML3是由OGC和ISO TC211发布的空间数据交换和编码的可扩展国际标准。CityGML基于ISO 191xx系列、OGC、万维网联盟（W3C）、Web3D联盟(W3D)和结构化信息标准推进组织（OASIS）的多种国际标准。

CityGML定义了城市和区域模型中最相关的对象类别及之间的关系，涉及到其几何、拓扑、语义和外观属性。“城市”的定义很宽泛，不仅包括建筑结构，还包括高程、植被、水体、城市家具等等。CityGML也包括了主题类、聚合、对象之间的关系和空间属性之间的细节层次模型。CityGML同时适用于大区域和小区域，可以表示不同细节层次的地形和三维对象。由于CityGML可以表示没有拓扑和语义的单一简单模型，也可以表示具有完整拓扑和细粒度语义差异的复杂多尺度模型，因此CityGML可以实现不同地理信息系统和用户之间的信息无损交换。

0.2. 历史背景

CityGML自2002年起由SIG 3D的成员开发。自2010年以后，该小组成为GDI-DE的一部分。2010年之前，SIG 3D隶属于北莱茵-威斯特伐利亚地理数据基础设施计划（GDI-NRW）。SIG 3D是一个开放的小组，由来自德国、英国、瑞士和奥地利的70多家公司、政府部门和研究机构组成，致力于开发可操作的三维城市模型和地理可视化。SIG 3D工作的另一个工作成果是提出了网络三维服务（W3DS），这是一个3D绘制服务，也是OGC正在讨论的（OGC Doc. No. 05-019 and OGC Doc. No. 09-104r1）。

2005年，GDI-NRW的 **Pilot 3D** 项目首次成功实施并评估了CityGML的一部分内容。与会者来自德国各地，展示了CityGML在城市规划方案和旅游上的应用。2006年初，EuroSDR的CityGML项目也开始关注欧洲的三维城市建模协调。2006年6月至12月，CityGML在OWS-4的CAD/GIS/BIM专题中得到应用和评估。自2008年以来，CityGML（1.0.0版本）成为OGC的一个标准。

从这个时间节点开始，CityGML已经在全世界传播。德国和欧洲其他国家的城市使用CityGML提供了各自的三维城市模型（仅举几个例子，柏林、科隆、德累斯顿和慕尼黑）。在法国，Bâti3D项目（IGN France）定义了CityGML LOD2的一个专用文件，并提供了巴黎和普罗旺斯艾克斯、里尔、南特和马赛市中心的数据。CityGML在荷兰的三维试点项目中也发挥了重要作用，支撑其三维地理信息标准和三维基础设施。欧洲的许多城市，如摩纳哥、日内瓦、苏黎世、利沃登，丹麦的城市（LOD2和3，部分是LOD4）都使用CityGML LOD2或3来进行数据的表达与交换。CityGML对欧盟委员会INSPIRE倡议的建筑模型（2.0版本）产生了重大影响，该

倡议旨在创建一个欧洲空间数据基础设施，以互操作方式提供公共数据。在亚洲，伊斯坦布尔（LOD1和LOD2）、多哈、卡塔尔（LOD3）和横滨（LOD2）的三维城市模型也使用CityGML进行数据的表达与交换。此外，CityGML在马来西亚的三维空间数据基础设施中起着至关重要的作用。

今天，许多商业和学术工具通过接口的方式来支持CityGML的使用。其中典型案例是三维城市数据库（3D City Database），该数据库是一个免费的开源3D地理数据库，主要用于存储、表示和管理由柏林理工大学提供的Oracle 10g R2和11g R1/R2+的虚拟三维城市模型。它完全支持CityGML，并附带了一个用于导入和导出CityGML模型的工具。此外，柏林技术大学还提供了一个用于处理CityGML模型（citygml4j）的开源Java类库和API。Safe软件公司的转换工具FME（要素操作引擎）是ESRI的ArcGIS交互性扩展的一部分，它也具有CityGML的读写接口。同样，Bentley Systems的BentleyMap等CAD工具，以及CPA Geo Information的SupportGIS等GIS工具也提供CityGML的读写接口。目前部分3D查看器（都是免费提供的）也为CityGML提供读取接口。例如，波恩大学的Aristoteles Viewer、Autodesk 的 LandXplorer CityGML Viewer（studio版本非免费）、KIT Karlsruhe的FZKViewer for IFC and CityGML以及Bitmanagement Software GmbH的BS Contact（通过地理空间扩展工具BS Contact Geo提供了CityGML插件）。关于CityGML转换工具正在日渐增多。如有需要，可访问CityGML的官方网站 http://www.citygml.org 以及 http://www.citygmlwiki.org。

0.3. CityGML 2.0 新功能

CityGML 2.0是针对CityGML国际标准1.0版本（OGC Doc. No. 08-007r1）的主要更新，为CityGML的主题模型引入了大量的新增内容和新功能。修订版最初计划是对1.1版的一个小更新。修订过程的主要努力是确保在概念模型成名和CityGML实例文档层面的向后兼容性。然而，某些更改无法按照 OGC政策（参见OGC文件 No.135r11）里关于小修订和向后兼容性的要求进一步实施。因此，考虑到CityGML社区的更新需求，将主要版本号更改为2.0，以符合OGC版本控制策略。

CityGML 2.0版本以下意义上向后兼容1.0版本：每个有效的1.0实例都是有效的2.0实例，前提是需要将文档中的CityGML命名空间和模式位置更改为对应的2.0名称，因为CityGML版本号编码已在源码中更新。

以下条款概述了CityGML2.0的新增功能。

新主题模块：桥梁和隧道

桥梁（*Bridge*）和隧道（*Tunnel*）是城市和景观模型中的重要对象。它们是交通基础设施的重要组成部分，通常是城市的标志性建筑。CityGML1.0一直缺乏专门用于桥梁和隧道的主题模块，因此必须使用 *GenericCityObject* 作为代理来建模和交换此类对象（参见第10.12章）。CityGML2.0现在引入了两个新的主题模块，明确表示桥梁和隧道。这是对CityGML主题模型的补充：桥梁模块（参见第10.4章）和隧道模块（参见第10.5章）。

桥梁和隧道可以用LOD1–4表示，基础数据模型与 *Building* 模型具有一致的结构。例如，可以将桥梁和隧道分解为多个部分，可以使用带有开口的主题边界面对壳体的各个部分进行语义分类，并且可以表示装置以及内部建造的结构。这种连贯的模型结构有助于对语义实体的理解，并有助于减少软件实现工作量。桥梁和隧道模型分别介绍了特定于桥梁和隧道的更多概念和模型元素。

对现有主题单元的补充

- *CityGML核心模块（参见第10.1章）*
  - 对抽象基类 *core:_CityObject* 添加了两个新的可选属性： *CityGML Core* 中的 *relativeToTerrain* 和 *relativeToWater* 。这些属性以定性的方式表示地物相对于地形和水面的位置，从而便于简单有效的查询（例如，查询地下建筑物的数量），不需要额外的数字地形模型或水体模型。
- *建筑物模块*
  - *LOD0 表示*
    - 建筑物现在可以使用LOD0对建筑底地面轮廓和/或屋顶边界进行表示。可以将现有的二维数据和来自航空和卫星图像的屋顶重建数据易于整合到一个三维城市模型中。目前仅限于水平的三维表面。
  - *附件的专题边界表面*
    - 为了从语义上，对既不是水平墙面也不是屋顶部分的建筑外壳进行分类，引入了两个附加的边界表面： *OuterFloorSurface* 和 *OuterCeilingSurface* 。
  - *与专题边界表面的附加关系*
    - 除了 *_AbstractBuilding* 和 *Room* ， *BuildingInstallation* 和 *IntBuildingInstallation* 的表面几何图形现在可以使用主题边界表面进行语义分类。例如，对于建模为 *BuildingInstallation* 的老虎窗（译注：即斜屋顶上的天窗），有助于对其屋顶和墙壁表面进行语义区分。
  - *隐式几何的附加使用*
    - 除了BuildingFurniture外，隐式几何图形（参见第8.3章）现在还可用于表示 *_Opening* 、 *BuildingInstallation* 和 *IntBuildingInstallation* 。使用隐式几何后，这些城市对象的原型可以在局部坐标系事先存储，并在三维城市模型的不同位置进行实例化。
- *泛型模块（参见第10.12章）*
  - 泛型模块中添加了两个泛型属性： *MeasureAttribute* 和 *GenericAttributeSet* 。 *MeasureAttribute* 有助于表示测量值以及对所用单位的引用。 *GenericAttributeSet* 是任意泛型属性的命名集合。它提供了一个codeSpace属性(可选)，用来表示定义该属性集的组织。
- *土地利用模块（参见第10.10章）*
  - *LandUse* 地貌类型的范围已经扩大，包括地球表面专门用于特定土地利用的区域和具有特定土地覆盖的区域，无论是否有植被。
- *属性类、函数和用法（所有模块）（参见第10.10章）*
  - 为了协调 *class* , *function* 和 *usage* 的使用，对于在CityGML 1.0中提供了一个及以上属性的所有要素类，都补充了该三元组。

对CityGML代码列表机制的补充

CityGML中，在代码列表里提供了类、函数和用法等枚举属性的数值。该代码列表可以由任何组织或社区根据其特定的信息需求在CityGML模式之外指定。然而，这种机制并没有完全反映在CityGML 1.0编码模式中，因为在CityGML 1.0实例文档中，相应的属性不能指向具有所用代码列表值的字典。这一点在CityGML 2.0中得到了纠正：所有从代码列表中获取值的属性都属于 *gml:CodeType* ，遵循GML 3.1.1机制对代码表值进行编码(参见第章10.14了解更多信息）。这个 *gml:CodeType* 为枚举属性添加了数值，该功能允许提供指向相应字典的URI。

CityGML 2.0的变更日志

附件F中提供了XML模式组件级别的更改。

对规范文档的进一步编辑

- 细节层次模型（LOD）的精度要求（参见第6.2章）
  第6.2章中提出的不同CityGML LOD精度要求是非规范性的。然而，CityGML 1.0中第6.2章的措辞与这一事实不一致，因此已针对CityGML 2.0进行了澄清。
- 修改CityGML示例数据集（参见附录G）
  附录G中提供的CityGML示例已被修改和扩展。现在，在示例中展示了用五种LOD等级表示的同一建筑模型，演示了不同LOD中建筑的语义和几何形态，以及使用XLinks在要素之间共享几何元素。数据集与CityGML XML模块包一起提供，可在以下网址中找到 http://schemas.opengis.net/citygml/examples/2.0/ 。
- 应用领域扩展使用的新示例（参见附件一）
  附件一中增加了泛在网络机器人服务领域，该案例为使用应用领域扩展的第二个例子。

1. 范围

本文档是OGC编码标准，用于虚拟3D城市和景观模型地表示、存储和交换。CityGML是基于GML3的一个具体应用模式加以实现。

CityGML主要对复杂且基于地理信息的三维矢量数据及其相关语义数据进行建模。与其他三维矢量格式相比，CityGML基于一个更加丰富的、通用的信息模型，该模型不仅仅包含几何和外观信息。对于特定领域，CityGML还提供了一种扩展机制，在保持语义操作性的前提下，用可识别的特征来丰富数据。

明确的可以应用的领域有：城市和景观规划、建筑设计、旅游和休闲活动、三维地籍、环境模拟、移动通信、灾害管理、国土安全、车辆和行人导航、培训模拟器和移动机器人。

CityGML被认为是3D绘制的源格式。模型中包含的语义信息可用于生成计算机图形（例如KML/COLLADA或X3D文件）的样式化过程。适用于此过程的OGC绘制网络服务（PWS）是OGC W3DS。OGC网络视图服务（WVS）为虚拟3D景观和城市模型提供基于图像的3D绘制服务。

CityGML具备以下特点：

- 基于ISO 191xx系列的城市景观地理空间信息模型（信息本体）
- 基于ISO 19107模型的3D几何图形GML3表示法
- 物体表面特征的表示（例如纹理、材料）
- 分类和聚合
    - 数字地形模型是由（包括嵌套的）不规则三角网（TINs）、规则栅格、断裂线和骨架线、质量点组成的
    - 场地（目前为建筑物、桥梁和隧道）
    - 植被（包含面积、体积和孤立对象）
    - 水体（体积、表面）
    - 交通设施（图形结构和三维表面数据）
    - 土地利用（具有不同特点的土地利用方式）
    - 城市家具
    - 通用的城市对象和属性
    - 用户可定义（迭代）分组
- 多尺度模型：具有5种明确定义的细节层次模型（LOD）
    - LOD0–区域，景观
    - LOD1–城市，区域
    - LOD2–城区，项目
    - LOD3–建筑模型（外部），地标
    - LOD4–建筑模型（内部）

- 不同LOD中的多重表示，不同LOD中对象之间的泛化关系
- 特征（子）几何体之间的可选拓扑连接
- 应用领域扩展（ADE）：CityGML模式中特定的部分，可用来定义特定的扩展程序，例如用于噪声污染模拟，或者通过美国新国家建筑信息模型标准（NBIMS）的属性来扩充CityGML。

2. 一致性

本国际标准所涉及的一致性目标仅适用于CityGML实例文件。本国际标准的未来修订也可能将消费者或生产者作为一致性目标。

本国际标准第8条至第10条规定了单独的CityGML XML模式定义和规范性内容（即CityGML模块），应根据第7条在CityGML实例文件中使用。本标准的实现不需要支持所有CityGML模块所提供的全部能力。根据第7.2章中CityGML profiles（专用文件）的规则和指南，部分模块和功能的有效实现是被支持的。

符合本国际标准的CityGML实例文件应具备以下三项条件：

- 符合第7章至第10章规定的规则和要求
- 通过附录B.1中抽象测试套件的所有相关测试案例
- 满足附录B.2中与CityGML模块有关的抽象测试套件中的所有一致性类别。

3. 规范性引用文件

以下规范性文件所包含的条款，通过在本文中的引用，构成了OGC 12-019的这一部分的规定。凡是标注日期的引用文件，其之后的修改或修订均不适用。但是鼓励根据OGC 12-019达成协议的各方研究，应用下列规范性文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

以下所列文件对于CityGML标准的应用是不可或缺的。除采用GML 3.1.1的几何模型之外，也使用了一些其它的附加概念，如隐式几何（参见第8.2章）、外观模型（参见第9章）借鉴了X3D和COLLADA的概念，以及地址使用OASIS可扩展地址语言xAL表示。

- ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*
- ISO/TS 19103:2005, *Geographic Information – Conceptual Schema Language*
- ISO 19105:2000, *Geographic information – Conformance and testing*
- ISO 19107:2003, *Geographic Information – Spatial Schema*
- ISO 19109:2005, *Geographic Information – Rules for Application Schemas*
- ISO 19111:2003, *Geographic information – Spatial referencing by coordinates*
- ISO 19115:2003, *Geographic Information – Metadata*
- ISO 19123:2005, *Geographic Information – Coverages*
- ISO/TS 19139:2007, *Geographic Information – Metadata – XML schema implementation*
- ISO/IEC 19775:2004, *X3D Abstract Specification*
- OpenGIS® Abstract Specification Topic 0, *Overview, OGC document 04-084*
- OpenGIS® Abstract Specification Topic 5, *The OpenGIS Feature, OGC document 99-105r2*
- OpenGIS® Abstract Specification Topic 8, *Relations between Features, OGC document 99-108r2*
- OpenGIS® Abstract Specification Topic 10, *Feature Collections, OGC document 99-110*
- OpenGIS® Geography Markup Language Implementation Specification, *Version 3.1.1, OGC document 03-105r1*
- OpenGIS® GML 3.1.1 Simple Dictionary Profile, *Version 1.0.0, OGC document 05-099r2*
- IETF RFC 2045 & 2046, *Multipurpose Internet Mail Extensions (MIME). (November 1996)*
- IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax. (August 1998)*
- W3C XLink, *XML Linking Language (XLink) Version 1.0. W3C Recommendation (27 June 2001)*
- W3C XMLName, *Namespaces in XML. W3C Recommendation (14 January 1999)*
- W3C XMLSchema-1, *XML Schema Part 1: Structures. W3C Recommendation (2 May 2001)*
- W3C XMLSchema-2, *XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001)*
- W3C XPointer, *XML Pointer Language (XPointer) Version 1.0. W3C Working Draft (16 August 2002)*

- W3C XML Base, *XML Base, W3C Recommendation (27 June 2001)*
- W3C XML, *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation (6 October 2000)*
- OASIS (Organization for the Advancement of Structured Information Standards): extensible Address Language (xAL v2.0).
- Khronos Group Inc.: COLLADA – Digital Asset Schema Release 1.4.1
- The Schematron Assertion Language 1.5. Rick Jelliffe 2002-10-01

4. 约定

4.1. 术语缩写

本文中使用的缩写如下：

缩写	英文全称	中文对应解释
2D	Two Dimensional	二维
3D	Three Dimensional	三维
AEC	Architecture, Engineering, Construction	建筑、工程、施工
ALKIS	German National Standard for Cadastral Information	德国地籍信息国家标准
ATKIS	German National Standard for Topographic and Cartographic Information	德国地形和制图信息国家标准
B-Rep	Boundary Representation	边界表示法
bSI	buildingSMART International	国际智慧建筑物联盟
CAD	Computer Aided Design	计算机辅助设计
COLLADA	Collaborative Design Activity	协同设计活动
CSG	Constructive Solid Geometry	构造实体几何
DTM	Digital Terrain Model	数字地形模型
DXF	Drawing Exchange Format	图形交换格式
EuroSDR	European Spatial Data Research Organisation	欧洲空间数据研究组织
ESRI	Environmental Systems Research Institute	环境系统研究所
FM	Facility Management	设施管理
GDF	Geographic Data Files	地理数据文件
GDI-DE	Spatial Data Infrastructure Germany (Geodateninfrastruktur Deutschland)	德国空间数据基础设施
GDI NRW	Geodata Infrastructure North-Rhine Westphalia	北莱茵河威斯特伐利亚地理数据基础设施

缩写	英文全称	中文对应解释
GML	Geography Markup Language	地理标记语言
IAI	International Alliance for Interoperability (now buildingSMART International (bSI))	国际互操作联盟（现已更名为国际智慧建筑物联盟）
IETF	Internet Engineering Task Force	互联网工程特别工作组
IFC	Industry Foundation Classes	工业基础类
ISO	International Organization for Standardisation	国际标准化组织
LOD	Level of Detail	细节层次模型
NBIMS	National Building Information Model Standard	国家建筑信息模型标准
OASIS	Organisation for the Advancement of Structured Information Standards	国际结构化信息标准推进组织
OGC	Open Geospatial Consortium	国际开放地理信息协会
OSCRE	Open Standards Consortium for Real Estate	房地产开放标准联盟
SIG 3D	Special Interest Group 3D of the GDI-DE	GDI-DE的三维特别兴趣小组
TC211	ISO Technical Committee 211	ISO 211技术委员会
TIC	Terrain Intersection Curve	地形相交曲线
TIN	Triangulated Irregular Network	不规则三角网
UML	Unified Modeling Language	统一建模语言
URI	Uniform Resource Identifier	统一资源标识符
VRML	Virtual Reality Modeling Language	虚拟现实建模语言

缩写	英文全称	中文对应解释
W3C	World Wide Web Consortium	万维网联盟
W3DS	OGC Web 3D Service	OGC网络三维服务
WFS	OGC Web Feature Service	OGC网络要素服务
X3D	Open Standards XML-enabled 3D file format of the Web 3D Consortium	Web3D联盟支持XML的3D文件开放标准
XML	Extensible Markup Language	可扩展标记语言
xAL	OASIS extensible Address Language	OASIS可扩展地址语言

4.2. UML表示法

在本文档中，使用统一建模语言（UML）静态结构图（见Booch et al. 1997）形式呈现CityGML标准，使用的UML表示法如下图1所示。

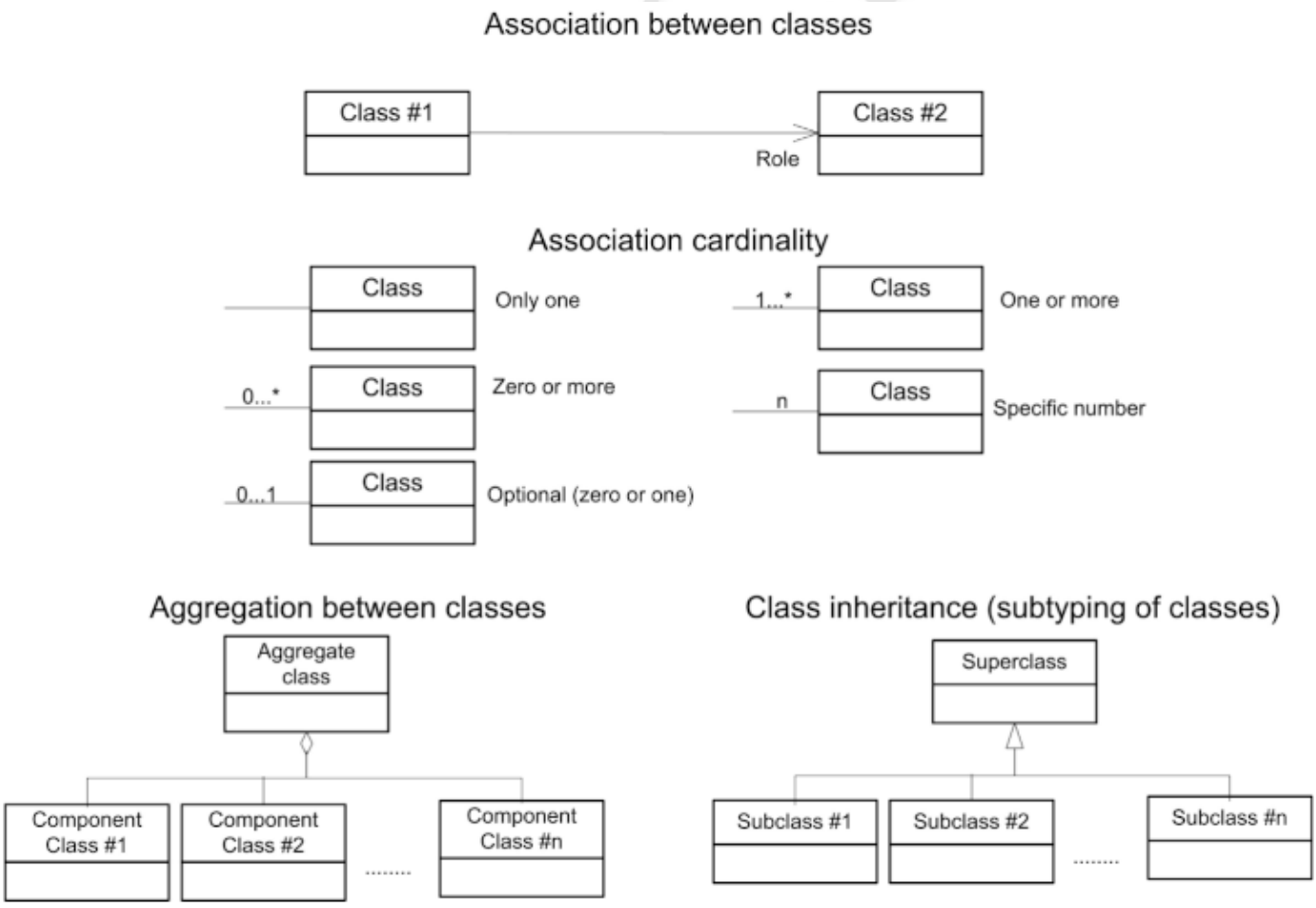

图 1. UML符号图（详见ISO TS 19103，地理信息-概念模式语言）

根据GML3,
CityGML中所有模型元素间的关联均为单向的。因此，CityGML中的关联只能单向导航。导航的方向以箭头表示。通常，关联中的元素，所使用的上下文由其角色进行指定。角色会显示在关联目标的附近。如果图形表示不明确，则必须将角色的位置绘制到关联所指向的元素上。

使用到的模板如下：

- «Geometry»表示对象的几何形状。几何图形是从抽象GML类型 *AbstractGeometryType* 派生的可识别和可区分对象。
- «Feature»表示根据ISO 19109定义的主题要素。要素是从抽象GML类型 *AbstractFeatureType* 派生的可识别和可区分对象。
- «Object»表示从抽象GML类型 *AbstractGMLType* 派生的可识别和可区分对象。
- «Enumeration»在指定文字值的固定列表中，枚举有效的属性值。枚举在CityGML模式中由内联指定。
- «CodeList»枚举有效的属性值。与枚举不同，值列表是开放的，因此在CityGML模式中没有给出。可以在外部代码列表中提供允许的值。建议按照GML3.1.1简单字典配置文件(参见第6.6章和第10.14章)将代码列表实现为简单字典。
- «Union»是一个属性列表。其语义是，在任何时候只能使用一个一个属性。
- «PrimitiveType»作为实现中原始类型支持的表示。
- «DataType»作为一组缺乏标识符的值的描述符。数据类型包括原始预定义类型和用户可定义类型。因此，数据类型是只有很少或无操作的类别，其主要目的是保存另一个类的抽象状态，用于传输、存储、编码或长期存储。
- «Leaf»是在UML包图中使用，表示没有其他子类型。
- «XSDSchema»在UML包图中使用，表示包含特定命名空间的所有定义的XSD模式的根元素。所有包内容或组件类都放在一个模式中。
- «ApplicationSchema»在UML包图中使用，表示XML模式定义从根本上依赖于XML模式元语言中另一个独立标准概念。例如，ApplicationSchema表示GML的扩展符合GML的“应用模式规则”。

为了增强CityGMLUML图表的可读性，如果类属于不同的UML包，则用不同的颜色描述（见图8 UML包的概述）。配色方案如下:

- 标为黄色的类 属于UML包讨论的主题。例如，在介绍 *CityGML Core* 模块的10.1章中，黄色用来表示 *CityGML Core* UML包中定义的类。同样，在10.3章UML图中显示的黄色类是该章中 *Building* 模块讨论的主题。
- 标为蓝色的类 与黄色相关联的包不同。为明确表示这些类，它们的类名称带有一个命名空间前缀，在整个规范中与CityGML模块唯一相关联（参见4.3节中命名空间和前缀列表）。例如，在 *Building* 模块的上下文中，*CityGML Core* 模块中的类被标为蓝色，它们的类名前面有前缀 *core* 。
- 标为绿色的类 是在GML3中定义的，它们的类名前面有 *gml* 前缀。

以下的UML图表示例展示了本规范中使用的UML表示法和配色方案。在本例中，黄色类与CityGML建筑模块相关联，蓝色类是 *CityGML Core* 模块，绿色类表示GML3定义的几何元素。

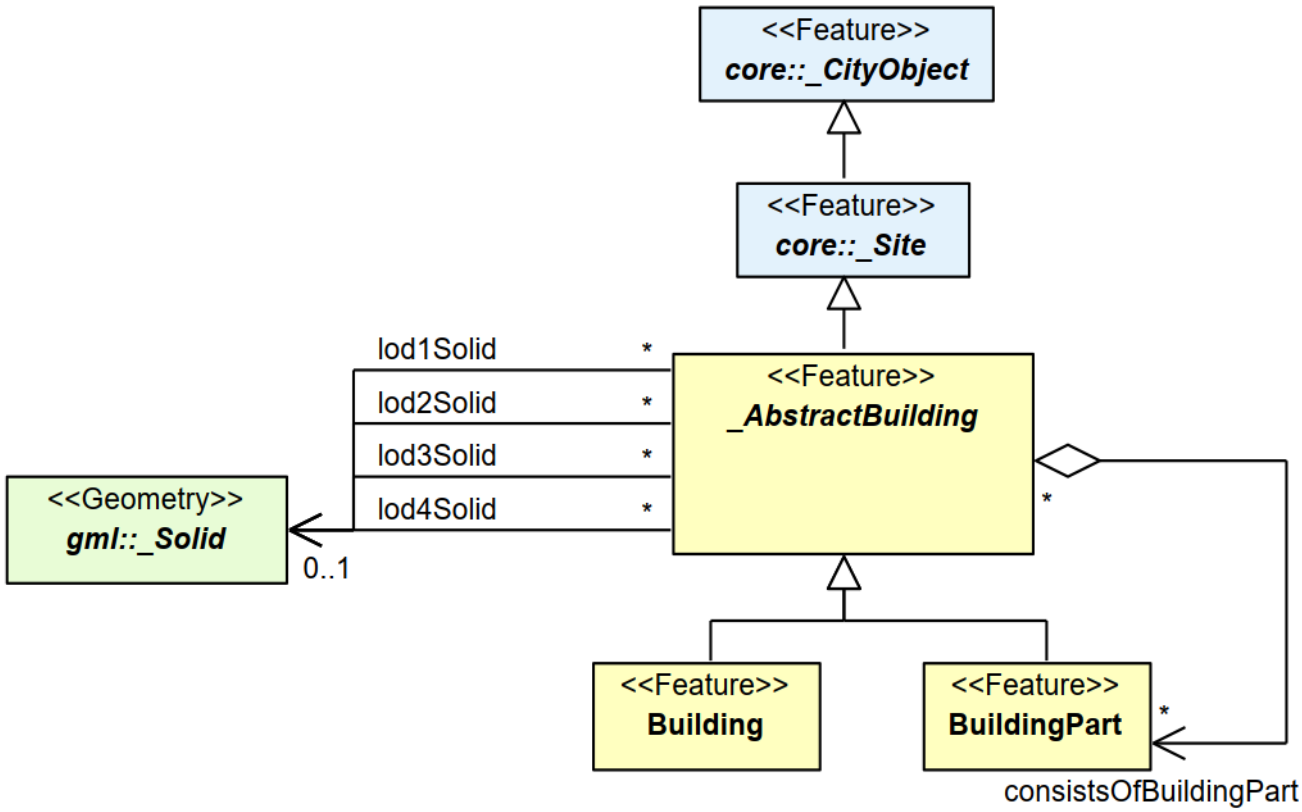

图 2. 展示了整个CityGML规范中使用的UML符号和配色方案的UML示例图

4.3. XML命名空间和命名空间前缀

CityGML数据模型按主题分解为核心模块和主题扩展模块。包括核心在内的所有模块都由其自身的XML模式文件指定，每个都定义了全局唯一的XML命名空间。扩展模块基于核心模块，因此包含（通过参考）CityGML核心模式。

在本文档中，模块命名空间与推荐的前缀相关联。这些前缀在本规范的规范性部分中统一使用，用于所有UML图和CityGML实例文档示例。表1中列出了CityGML核心模块和扩展模块及其XML命名空间标识符和推荐的命名空间前缀。

表1: CityGML模块列表，其关联的XML命名空间标识符，以及命名空间前缀示例

CityGML模块	命名空间标识符	命名空间前缀
CityGML核心	http://www.opengis.net/citygml/2.0	core
外观	http://www.opengis.net/citygml/appearance/2.0	app
桥梁	http://www.opengis.net/citygml/bridge/2.0	brid
建筑	http://www.opengis.net/citygml/building/2.0	bldg

CityGML模块	命名空间标识符	命名空间前缀
城市家具	http://www.opengis.net/citygml/cityfurniture/2.0	frn
城市对象群组	http://www.opengis.net/citygml/cityobjectgroup/2.0	grp
泛型	http://www.opengis.net/citygml/generics/2.0	gen
土地利用	http://www.opengis.net/citygml/landuse/2.0	luse
地形	http://www.opengis.net/citygml/relief/2.0	dem
交通设施	http://www.opengis.net/citygml/transportation/2.0	tran
隧道	http://www.opengis.net/citygml/tunnel/2.0	tun
植被	http://www.opengis.net/citygml/vegetation/2.0	veg
水体	http://www.opengis.net/citygml/waterbody/2.0	wtr
纹理表面 [已弃用]	http://www.opengis.net/citygml/texturedsurface/2.0	tex

表2中显示了与该标准相关的XML模式定义，以及本文档中一致使用的相应XML命名空间标识符和命名空间前缀。

表2: XML模式定义列表、其相关的XML命名空间标识符，以及本文档中使用的命名空间前缀示例

XML模式定义	命名空间标识符	命名空间前缀
地理标记语言3.1.1版(来自OGC)	http://www.opengis.net/gml	gml
可扩展地址语言2.0版(来自OASIS)	urn:oasis:names:tc:ciq:xsdschema:xAL:2.0	xAL
Schematron Assertion语言1.5版	http://www.ascc.net/xml/schematron	sch

4.4. XML模式

该标准的规范部分使用W3C XML模式语言，来描述符合CityGML数据实例的语法。XML模式是一种具有多种功能的语言。虽然不熟悉XML模式的读者可以按照一般的方式查看描述，但该标准并非用于介绍XML模式。为了充分

理解该标准，读者需要对XML模式有一定的了解。

5. CityGML概述

CityGML是一种基于XML格式的、用于存储和交换虚拟3D城市模型的开放数据模型。它是地理标记语言3.1.1版本(GML3)的一个应用模式，是由OGC和ISO TC211发布的用于空间数据交换的可扩展国际标准

开发CityGML的目标是为了对三维城市模型的基本实体、属性和关系达成一个共同的定义。这对于三维城市模型的低成本、可持续性维护尤其重要，因为其允许在不同的应用领域重复使用相同的数据。

CityGML不仅表现了城市模型的图形外观，而且特别解决了语义和主题属性、分类法和聚合的表示。CityGML包括一个几何图形模型和一个主题模型。CityGML包括一个几何模型和一个主题模型。几何模型允许对三维城市模型中的空间对象的几何和拓扑属性进行一致和同质的定义（第8章）。所有对象的基类是 *_CityObject*，它是GML的 *_Feature*类的一个子类。所有的对象都继承了 *_CityObject*的属性。

CityGML的主题模型采用了不同主题领域的几何模型，如数字地形模型、场地（即建筑物、桥梁和隧道）、植被（单独的对象以及区域和体积的生物群落）、土地利用、水体、交通设施和城市家具（第10章）。更多尚未明确建模的对象，可以使用通用对象和属性的概念来表示（第6.11章）。此外，对CityGML数据模型的扩展，适用于特定的应用领域，可以使用应用域扩展（ADE）来实现（第6.12章）。相同形状的空间对象，如树等，在不同的位置出现多次，也可以被建模为原型，并在城市模型中多次使用(第8.2章)。分组的概念可以将单个三维物体组合起来，例如将建筑物组合成一个建筑群（6.8章）。没有用封闭实体进行几何建模的对象，可以使用*ClosureSurfaces*封闭(第6.4章)被虚拟密封，以计算其体积（例如人行地下通道、隧道或飞机库）。*TerrainIntersectionCurve*的概念被引入，将三维物体与数字地形模型的正确位置结合起来，以防止建筑物漂浮或沉入地形（6.5章）。

CityGML区分了五个连续的细节层次模型(LOD)，对象的几何图形和主题差异随着LOD的增加而变得更加详细(第6.2章)。CityGML文件可以-但不是必须-同时为不同LOD中的每个对象包含多种表示(和几何图形)。泛化关系允许在不同尺度上明确表示聚合对象。

除了空间属性之外，CityGML的特征还可以被赋予外观。外观并不局限于视觉数据，而是代表地物表面的任意可观察的属性，如红外辐射、噪声污染或地震引起的结构应力（第9章）。

此外，对象可以是对外部数据集中相应对象的外部引用（第6.7章）。枚举型对象属性的可能属性值，可以在外部可重新定义的字典中定义的代码列表中枚举（第6.6章）。

6. CityGML一般性特征

6.1. 模块化

CityGML数据模型由虚拟三维城市模型中最重要的对象类型的类定义组成，这些类在许多不同的应用领域都被认为是必需的或重要的。然而，我们并不需要为了符合标准而去支持整个CityGML数据模型，而是根据实际需求构造一个CityGML数据模型的子集即可。为此，CityGML数据模型采用了模块化结构(参见第7章)。

CityGML数据模型按主题分解为核心模块和主题扩展模块。核心模块包括CityGML数据模型的基本概念和组件，它是任何遵循标准的系统必须实现的。基于核心模块，每个扩展模块涵盖了虚拟三维城市模型的一个特定主题领域。CityGML介绍了以下13个主题扩展模块:外观、桥梁、建筑、城市家具、城市对象组、泛型城市对象、土地土地利用、地形、交通、隧道、植被、水体和纹理表面[已弃用]。

与核心模块相结合的任何扩展模块的任意组合，均是符合CityGML标准的实现。这种模块组合称为CityGML profiles(CityGML专用文件)。因此，CityGML专用文件允许对整个CityGML数据模型进行有效的部分实现。

6.2. 多尺度建模（5级LOD）

CityGML支持不同级别的LOD。LODs反映了具有不同应用需求的独立数据收集过程。此外，LOD有助于高效可视化和数据分析(如图3所示)。在一个CityGML数据集中，同一物体可以同时用不同的LOD来表示，这样就可以在不同的分辨率下对同一个物体进行分析和可视化。此外，两个包含有不同LOD的同一物体的CityGML数据集可以被合并和整合。然而，用户或应用程序有责任确保不同LOD的物体是指同一个真实世界的物体。

最粗略的LOD0级别基本上是一个2.5维的数字地形模型，上面覆盖着航空图像或地图。在LOD0中，建筑可以用迹线或屋顶边缘多边形来表示。众所周知，LOD1是由带平屋顶结构的棱柱状建筑组成的块状模型。相比之下，LOD2中的建筑有不同的屋顶结构和不同主题的边界表面。LOD3表示有详细的墙和屋顶结构的建筑模型，可能包含门和窗户。LOD4在LOD3的基础上又为建筑物添加了内部结构。例如，LOD4中的建筑由房间、室内门、楼梯和家具组成。在所有的LOD外观信息，如高分辨率纹理可以映射到结构中(参见6.9)。

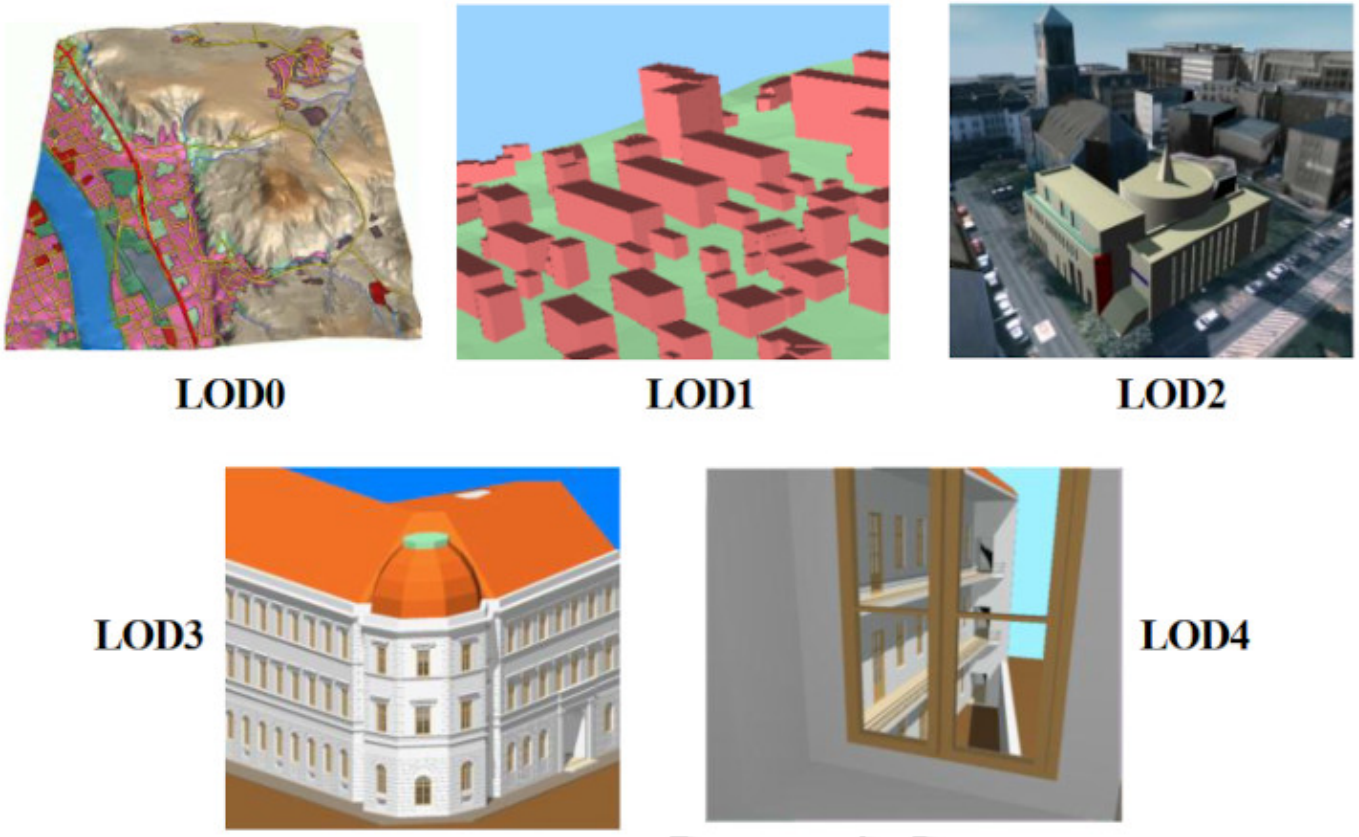

图 3. CityGML定义的5个LOD等级（来源于IGG Uni BOom）

LOD的特点还包括不同的精度和物体的最小尺寸(参见表3)。本标准中给出的精度要求尚未最终确定，仅作为讨论建议。精度描述为绝对三维点坐标的标准差。在未来版本的CityGML中将增加相对三维点精度，它通常比绝对精度要高得多。在LOD1中，点的定位精度和高度精度应小于等于5m，而所有的对象都应至少为6m × 6m的迹线。LOD2的定位精度和高度精度建议在2m以上，在此LOD中，所有迹线至少是4m × 4m的对象都应该被考虑。LOD3中两种类型的精度都应该是0.5m，建议最小迹线为2m × 2m。最后，LOD4的定位精度和高度精度应优于0.2m。通过这些数据，5个LOD的分类可以用来评估三维城市模型数据集的质量，并使数据集具有可比性，为其整合提供支持。

表3: CityGML的LOD 0-4及其提出的精度要求(讨论方案，基于: Albert 发布于2003年)。

	LOD0	LOD1	LOD2	LOD3	LOD4
模型尺度描述	区域性景观	城市, 区域	市区, 规划项目	建筑模型 (外观), 地标	建筑模型 (内部)
分类精度	最低	低	中等	高	很高
绝对 3D 点精度（位置/高度）	低于 LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
概化程度	最大概化（土地利用分类）	具有抽象特征的对象块; >6*6m/3m	具有抽象特征的对象; > 4*4m/2m	具有真实特征的对象; > 2*2m/1m	建筑要素和出入口被表达
建筑物设施	-	-	-	有代表性的外部效果	实物形态
屋顶样式/结构	无	平面	屋顶类型和方向	实物形态	实物形态
屋顶悬梁部分	-	-	无	无	有
城市家具	-	重要对象	原型样板	实物形态	实物形态
孤立植被对象	-	重要对象	原型样板, 高于 6m	原型样板, 高于 2m	原型样板, 实物形态
土地利用	-	>50*50m	>5*5m	< LOD2	<LOD2
… 其他功能主题					

而在CityGML中，每个对象在每个LOD中都可以有不同的表示，通常同一LOD中的不同对象会被概括为较低LOD中的一个聚合对象来表示。CityGML通过在城市对象之间提供一个明确的概括关联来支持聚合/分解（进一步的细节见第10.1章的UML图）。

6.3. 连贯语义几何建模

CityGML最重要的设计原则之一是语义和几何/拓扑属性的一致性建模。在语义层，真实世界的实体由要素表示，如建筑、墙壁、窗户或房间。描述还包括要素间的属性、关系和聚合层次结构(部分-整体-关系)。因此，要素之间的部分关系只能在语义层得到，无需考虑几何图形。但是，在空间层面上，几何对象被分配到表示其空间位置和范围的要素中。因此，该模型由两个层次结构组成:语义层和几何层，在这两个层中，相应的对象通过关系关联在一起(参考Stadler & Kolbe 2007)。这种方法的优点是，在语义/或几何查询或执行分析中，可以在语义和几何的两个层次结构中以及在两个层之间任意对象间引用。 如果两个层次都是针对一个特定对象而存在，那么它们必须是一致的(即必须确保它们互相匹配)。例如，如果建筑的墙壁在语义层上有两个窗户和一个门，那么表示墙壁的几何图形也必须包含窗户和门的几何部分。

6.4. 闭合曲面

没有体积几何模型的物体，必须使用*ClosureSurface*被虚拟封闭以计算其体积（例如人行地下通道或飞机库）。一共有13个特殊的表面，当需要计算体积时，它们会被考虑进去，而当它们不相关或不合适时，例如在可视化中，就会被忽略。

闭合面的概念也被用来对地下物体的入口进行建模。隧道或人行地下通道等物体必须被建模成封闭的实体，用于计算其体积，例如在洪水模拟。地下物体的入口也必须封闭，以避免数字地形模型中的洞(见图4)。但是，在近距离可视化中，入口必须开放。因此，闭合面是建模这些入口的合理方法。

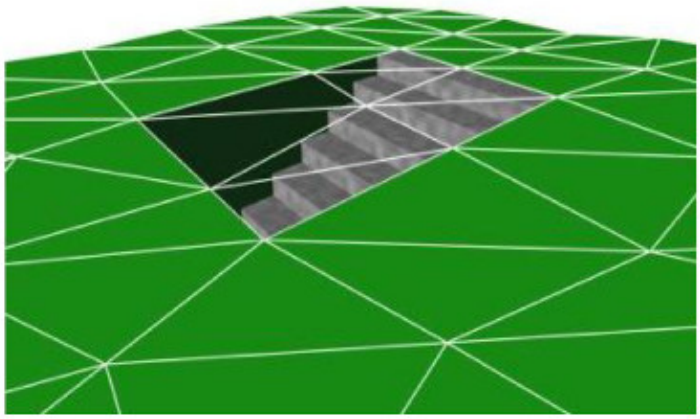

图 4. 闭合曲面封闭开放结构。通道是地下物体(左)。入口被虚拟闭合曲面封闭，它既是DTM的一部分，也是地下物体的一部分(右)(图:IGG Uni Bonn)。

6.5. 地形相交曲线（TIC）

城市建模的一个关键问题是三维物体和地形的整合。比如，将不同LOD的地形和三维物体结合起来，或者它们来自不同的供应商，经常会出现三维物体漂浮在地形上或沉入地形中这种问题（Kolbe和Gröger 2003）。为了克服这个问题，引入了三维物体的地形交汇曲线（*TerrainIntersectionCurve*）。这些曲线表示地形与三维物体接触的确切位置（见图5）。TIC可以应用于建筑物和建筑物部件（参见第10.3章）、桥梁、桥梁部件和桥梁建筑元素（参见第10.5章）、隧道和隧道部件（参见第10.4章）、城市家具对象（参见第10.9章）和一般城市对象（参见第10.12章）。例如，如果一个建筑物有一个庭院，那么TIC由两个封闭的环组成：一个环代表庭院的边界，另一个环描述建筑物的外部边界。这些信息可以用来整合建筑和地形，通过 "拉高 "或 "拉低 "周围的地形来适应地形交汇曲线，或者局部扭曲DTM以适应TIC。通过这种方式，TIC也确保了纹理的正确定位或物体纹理与DTM的匹配。由于不同LOD上与地形的相交点可能有所不同，一个三维对象对于所有LOD可能有不同的__TerrainIntersectionCurve_。

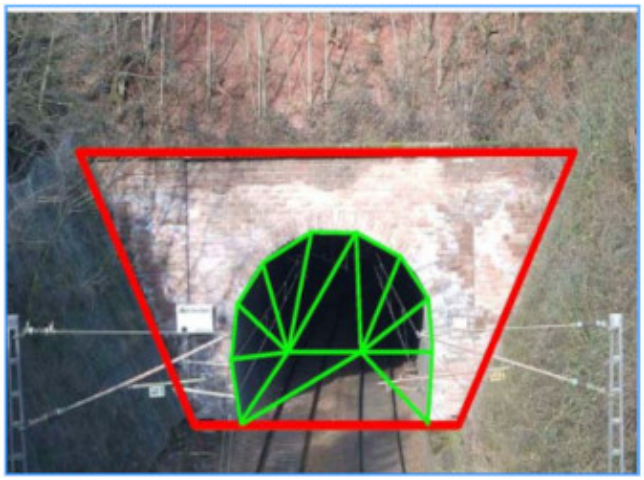

图 5. 建筑物（左边，黑色）和隧道物体（右边，白色）的地形交叉曲线。隧道的中空空间被一个三角形的ClosureSurface密封（图形：IGG Uni Bonn）。

6.6. 枚举属性代码表

CityGML要素类型通常包括一些属性，它们的值可以在离散值列表中枚举。例如，建筑的屋顶类型属性，其属性值通常为鞍形屋顶、四坡屋顶、半四坡屋顶、平屋顶、单斜面屋顶或帐篷屋顶。如果这样的属性作为字符串输入，则相同概念的拼写错误或不同名称会阻碍互操作性。此外，可能的属性值列表往往不是固定的，而且可能因不同的国家(例如，国家法律和条例规定)和不同的信息社区而有很大的不同。

在CityGML中，这些枚举属性的类型是*gml:CodeType*，其所允许的属性值可以在CityGML模式之外指定的代码列表中提供。一个代码列表包含了已编码的属性值，并确保相同的代码用于相同的概念或者框架。如果为一个枚举性属性提供了一个代码表，那么该属性只能从这个列表中取值。这允许应用程序验证属性值，从而促进语义和语法上的互操作性。我们建议按照GML 3.1.1简单字典的规定，将代码表作为简单字典来实现（参见Whiteside 2005）。

代码表的管理与CityGML模式和规范的管理是脱钩的。因此，代码表可以由任何组织或信息社区根据他们的信息需求来指定。每个代码表应该有一个权威机构，负责代码表的值和代码表的维护。关于CityGML代码表机制的进一步信息，见第10.14章。

代码列表可以参考现有的模型。例如，可以参考房地产开放标准联盟（OSCRE）定义的房间代码，也可以使用国家建筑信息模型标准（NBIMS）引入的建筑和建筑部分的分类方法。附件C包含了SIG 3D为CityGML中几乎所有的枚举性属性提出的非规范性代码列表。它们可以在CityGML实例文件中直接引用，并作为定义代码表的一个例子。

6.7. 外部引用

三维对象通常来自于其他数据库或数据集中的对象，或者与之有关系。例如，一个三维建筑模型可能是由地籍数据集中的二维迹线构建的，也可能是由建筑模型衍生出来的（图6）。如果必须传播更新或需要额外的数据，例如地籍信息系统中建筑物所有者的姓名和地址,或设施管理系统中的天线和门的信息，那么三维物体对其在外部数据集中的相应物体的引用是至关重要的。为了提供这些信息，每个 *_CityObject*可以使用*ExternalReference*的概念来引用外部数据集（UML图见图21；XML模式定义见附件A.1）。这种引用表示外部信息系统和该系统中对象的唯一标识符<两者都被指定为统一资源标识符（URI），这是对互联网上任何类型资源的引用的通用格式。外部引用的通用概念允许任何 *_CityObject*与外部信息系统中的相应对象（如ALKIS、ATKIS、OS MasterMap®、GDF等）有任意数量的链接。

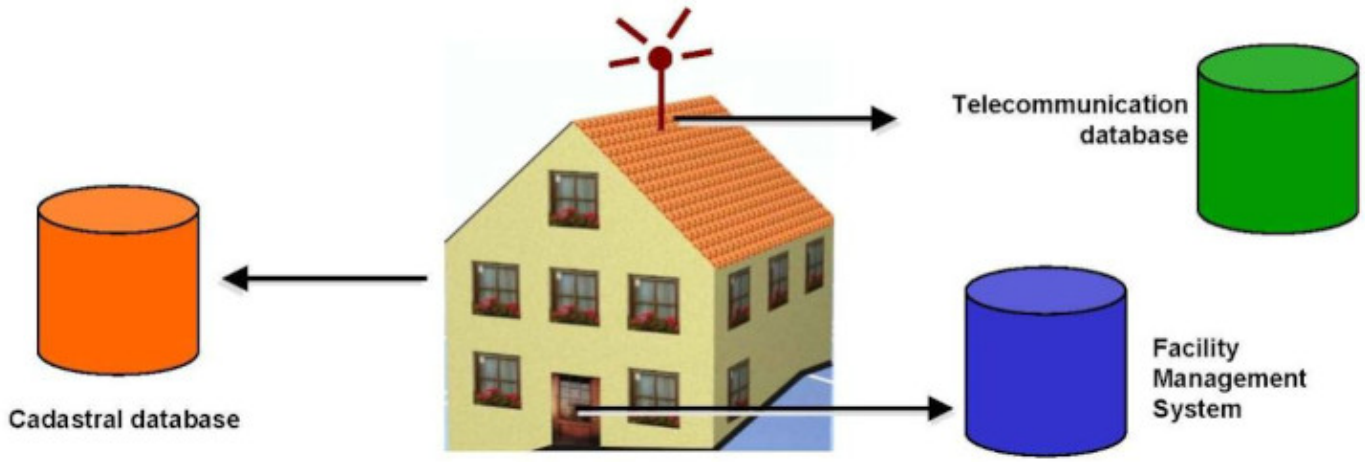

图 6. 外部引用（图片：IGG Uni Bonn）。

6.8. 城市对象组

CityGML的分组概念允许根据用户定义的标准对任意的城市对象进行聚合，并将这些聚合作为城市模型的一部分来表示和传输（UML图见OGC 12-019 Copyright © 2012 Open Geospatial Consortium. 15 第10.11章；XML模式定义见附件A.6）。) 一个组可以被赋予一个或多个名称，并可以通过特定的属性进一步分类，例如，"在一个火灾场景中，第43号房1212房间的逃生路线"作为名称，"逃生路线"作为类型。该组的每个成员可以选择被分配一个角色名称，该名称指定该特定成员在该组中扮演的角色，这个角色名称可以描述这个物体在逃生路线中的序列号，或者在一个建筑群的情况下表示主楼。

一个组可以包含其他组作为其成员，允许任意深度的嵌套分组。分组概念由CityGML的主题扩展模块*CityObjectGroup*提供(参见第10.11章)。

6.9. 外观

在虚拟三维城市模型中，表面的外观信息，即表面的可观察属性，被认为是除语义和几何图形之外的重要组成部分。外观涉及任何基于表面的主题信息，例如红外辐射或噪音污染，而不仅仅限于视觉属性。因此，外观提供的数据可以作为输入，用于在虚拟三维城市模型中进行展示和分析。

CityGML支持每个城市模型任意数量的主题的要素外观。每个要素的LOD可以有单独的外观。外观可以表示纹理和地理参考纹理。CityGML的外观模型被封装在自己的扩展模块*Appearance*中(参见第9章)。

6.10. 原型对象/场景图概念

在CityGML中，像树木、交通灯、及交通标志等形状相同的对象可以表示为原型，在不同的位置多次实例化（图7）。原型的几何形状是在本地坐标系中定义的。每个实例都由一个对原型的引用、世界坐标参考系中的一个基点和一个便于原型缩放、旋转和平移的变换矩阵来表示。这一原则来自于计算机图形标准（如VRML和X3D）中使用的场景图的概念。由于GML3几何模型没有提供对场景图概念的支持，它被作为GML3几何模型的一个扩展来实现（进一步的描述参见第8.2章）。

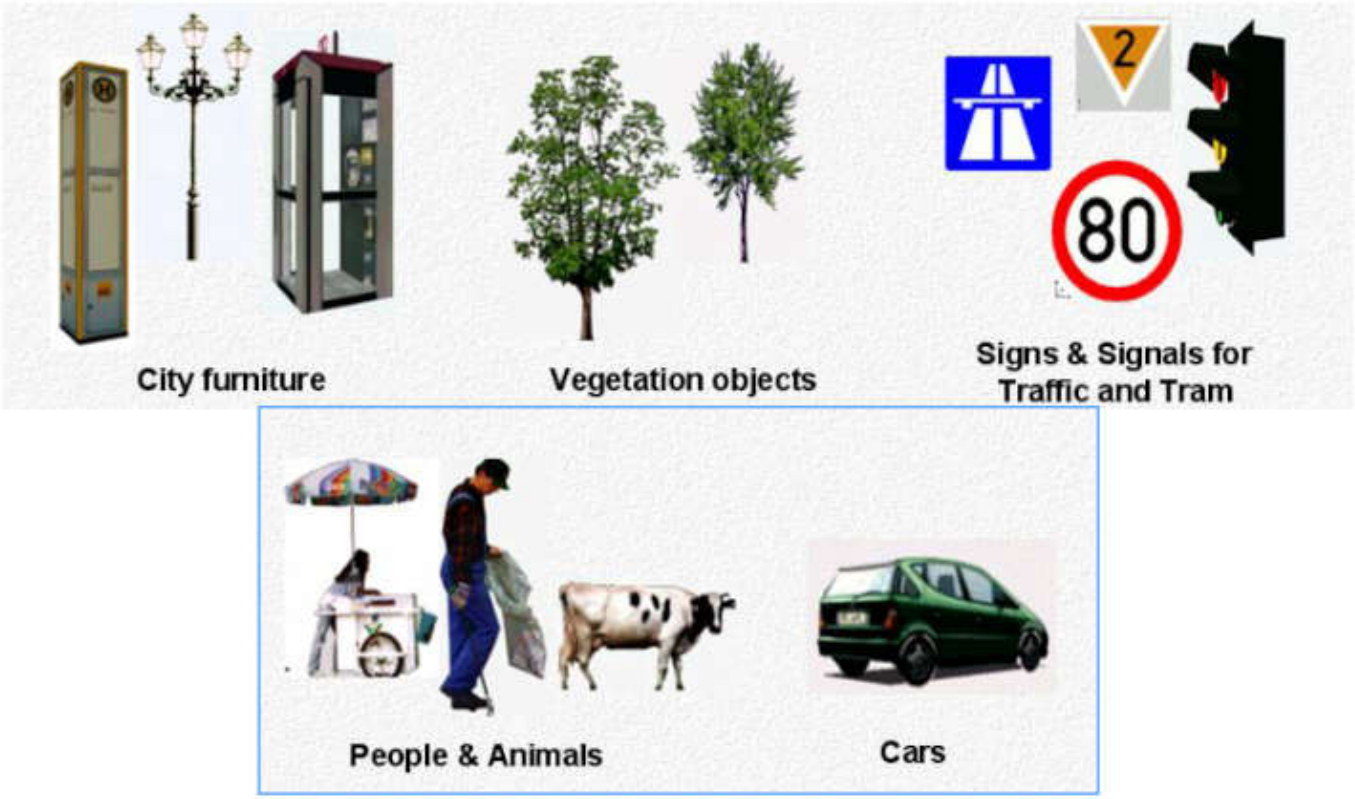

图 7. 原型形状的例子（来源：莱茵金属防御电子公司）。

6.11. 通用城市对象和属性

CityGML被设计成一个通用的地形信息模型，定义了对广泛的应用有用的对象类型和属性。在实际应用中，特定的三维城市模型中的对象很可能包含CityGML中没有明确建模的属性，此外，可能有一些三维物体没有被CityGML的主题类所覆盖。对此，CityGML提供了两种不同的概念来支持这些数据的交换。1）通用对象和属性，以及2）应用领域扩展（参见6.12章）。

通用对象和属性的概念允许在运行期间对CityGML应用程序进行扩展，即任何 *_CityObject_都可以由额外的属性来增加，这些属性的名称、数据类型和值可以由运行中的应用程序提供，而无需对CityGML的XML模式进行任何改变。同样，没有被CityGML数据模型的预定义主题类所代表的特征也可以使用通用对象进行建模和交换。CityGML的通用扩展是由主题扩展模块*Generics__提供的（参见第10.12章）。

CityGML的当前版本不包括诸如路堤、挖掘和城墙等明确主题模型，对于这些对象，可以使用通用对象和属性进行存储或交换。

6.12. 应用领域扩展（ADE）

应用领域扩展(ADE)是对CityGML数据模型的补充。这些补充包括在现有CityGML类中引入新属性，例如建筑的居民数量或新对象类型的定义。ADE与通用对象和属性之间的区别是，ADE必须被定义在一个额外的XML模式定义文件中，有它自己的命名空间。这个文件必须明确地导入扩展的CityGML模块的XML模式定义。

这种方法的优点是，扩展是正式规定的。扩展的CityGML实例文件可以根据CityGML和各自的ADE模式进行验证。ADEs可以由对特定应用领域感兴趣的信息团体来定义（甚至是标准化）。在同一个数据集中可以积极使用一个以上的ADE（进一步描述见第10.13章）。

ADE可以为一个甚至几个CityGML模块定义，为CityGML数据模型添加附加信息提供了高度的灵活性。因此，ADE机制是与CityGML的模块化方法正交的。因此，没有单独的ADE的扩展模块。

在本规范中，包括了两个ADE的例子。

- 噪声排放模拟的ADE（附件H）：根据欧盟委员会的环境噪声指令（2002/49/EC），用于模拟环境噪声的扩散；
- 泛在网络机器人服务的ADE（附件I）：展示了CityGML在室内环境中机器人导航的使用。

ADE的其他例子有：用于设施管理的*CAFM ADE*（Bleifuß等人，2009）、用于多设施网络及其相互依存关系的综合三维建模的*UtilityNet- workADE*（Becker等人，2011）、用于水文应用的*HydroADE*（Schulte和Coors，2008）以及*GeoBIM（IFC）ADE*（van Berlo等人，2011），它结合了IFC（来自BSI）和CityGML的BIM信息，并在开源模型服务器BIMserver.org中实施。

7. 模块化

CityGML在其数据模型的主题和几何拓扑级别上都是一个丰富的标准。在其主题级别上，CityGML定义了城市中最相关的地形对象的类和关系，以及包含建筑结构、高程、植被、水体、城市家具等的区域模型。除了几何和外观内容之外，这些主题组件还允许在不同的应用领域(如模拟、城市数据挖掘、设施管理和主题查询)中使用虚拟3D城市模型来完成复杂的分析任务。

CityGML是一个框架，为地理空间三维数据提供了足够的空间，让其在生命周期内可以在几何、拓扑和语义方面增长。因此，城市对象的几何和语义可以被灵活地结构化，从纯粹的几何数据集到复杂的有几何拓扑和空间语义的数据。通过这种方式，CityGML定义了一个适用于3D城市建模的连续过程、具有单一的对象模型和数据交换的格式，建模过程从几何数据获取、数据分类到特定终端用户应用的数据准备，允许迭代数据丰富和无损信息交换(cf. Kolbe et al. 2009)。

基于这个框架，应用程序不需要支持所有的CityGML主题字段以符合标准，但可以使用与应用领域或过程步骤相关需求对应的构造子集。CityGML逻辑子集的使用限制了整体数据模型的复杂性，并明确允许实现有效部分。对于CityGML标准的2.0版本，CityGML模块定义并包含了数据模型的子集。CityGML模块是标准的集合，由一个符合标准的系统作为整体。CityGML由一个核心模块和若干主题扩展模块组成。

CityGML核心模块定义了CityGML数据模型的基本概念和组件。它被看作是整个CityGML数据模型的通用下限，是所有主题扩展模块的基础。因此，核心模块是唯一的，且任何符合要求的系统都必须实现。基于CityGML核心模块，每个扩展模块都包含一个逻辑上独立的CityGML数据模型主题组件。核心的扩展是通过对整个CityGML数据模型进行垂直切片而得到的。由于每个扩展模块都包含（或引用）核心模块，因此它的概念和组件对所有扩展模块都是通用的。CityGML标准2.0版引入了以下13个主题扩展模块。它们与本文档的条款相关，每个条款都涵盖了CityGML相应的主题领域：

- 外观（参见第9条），
- 桥梁（参见第10.5条），
- 建筑 (参见第10.3条），
- 城市家具（参见第10.9条），
- 城市对象组合（参见第10.11条），
- 通用（参见第10.12条），
- 土地利用（参见第10.10条），
- 地形（参见第10.2条），
- 交通（参见第10.7条），
- 隧道（参见第10.4条），
- 植被（参见第10.8条），
- 水体（参阅第10.6条），和
- 带纹理的表面[已弃用]（参见第9.8条)

CityGML数据模型的按主题模块进行分解，允许实施者支持扩展模块与核心模块的任意组合。因此，扩展模块可以根据应用或应用领域的信息需求而任意组合。一个模块的组合称为CityGML专用文件（profiles）。所有模块

的联合体被定义为CityGML基本专用文件（base profile）。基本专用文件在任何时候都是唯一的，并形成了整个CityGML数据模型的上限。任何其他CityGML专用文件必须是基本专用文件的有效子集。通过遵循CityGML模块和专用文件，CityGML的部分数据模型以一种定义明确的方式实现是有效的。

至于未来的发展，每个CityGML模块都可以由专家小组和信息社区独立地进一步发展。由此产生的对模块的建议和更改可能会被引入到CityGML标准的修订中，而不会影响其它模块的有效性。此外，当前CityGML数据模型未覆盖的主题组件可以通过主题扩展模块添加到标准的未来版本中。这些扩展模块可与任何其它现有的CityGML模块建立依赖关系，但至少应依赖于CityGML核心模块。因此，随着新扩展模块的添加，CityGML基本专用文件可能也会变化。但是，如果一个特定的应用程序需要的建模和交换的信息超出了CityGML数据模型的范围，这些应用数据也可以使用CityGML的 *Application Domain Extension* 机制（参见第10.13条）或使用通用城市对象和属性（参见第10.12章）纳入现有模块中。

引入的模块化方法支持CityGML作为数据建模框架和通用数据交换格式，以解决各种应用领域和三维城市建模的不同步骤。为了清晰起见，应用程序应该通过声明所使用的CityGML专用文件来宣布与CityGML标准的一致性水平。由于核心模块是所有专用文件的一部分，这应该通过列举已实现的主题扩展模块来实现。例如，如果一个实现除了支持核心模块外，还支持 *Building*、 *Relief* 和 *Vegetation* 模块，则应该通过“CityGML [Building,Relief,Vegetation]”来表示。如果支持基本专用文件，则应该用“CityGML [full]”表示。

7.1. CityGML核心和扩展模块

每个CityGML模块都由自己的XML模式定义文件指定，并被定义在一个单独且全局唯一的XML目标命名空间中。根据模块之间的依赖关系，每个模块还可以导入与CityGML相关模块关联的命名空间。但是，一个命名空间不能直接包含在两个模块中。因此，所有属于一个模块的元素都只与该模块的命名空间相关联。通过这种方式，保证了模块元素在CityGML实例文档中被适当地分离和区分。

与1.0之前的CityGML版本相比，前面提到的命名空间约定给数据文件带来了额外的复杂性，因为不再有单独的CityGML命名空间。相反，不同CityGML模块的组件以及不同命名空间的组件可以任意混合在同一个CityGML实例文档中。此外，一个应用程序可能必须要解析包含模块元素的实例文档，而这些模块本身并不被应用程序所采用。不过，这些解析问题可以很容易地被非“模型感知”的应用程序克服，即那些不以通用方式解析和解释GML应用模式的应用程序。来自不同命名空间的与应用程序所使用的CityGML专用文件所声明的不同的元素，可能会被跳过。在使用CityGML的应用领域扩展机制（参见第10.13条）时，必须进行类似的观察。

CityGML标准的2.0版本，没有两个互相依赖的主题扩展模块，所有的扩展模块都是真正相互独立的。然而，CityGML核心模块是任何扩展模块的依赖项，这意味着定义扩展的每个XML模式文件都要导入核心模块的XML模式文件。

图8使用UML包图说明了CityGML模块之间的依赖关系。每个模块都由一个包表示，包名与模块名对应。图中的虚线箭头表示箭头尾部的模式依赖于箭头头部的模式。对于CityGML模块，当一个模式<import>另一个模式以及相应的XML命名空间时会产生依赖关系。例如，扩展模块建筑导入CityGML核心模块的模式。表4给出了每个模块的简短描述。

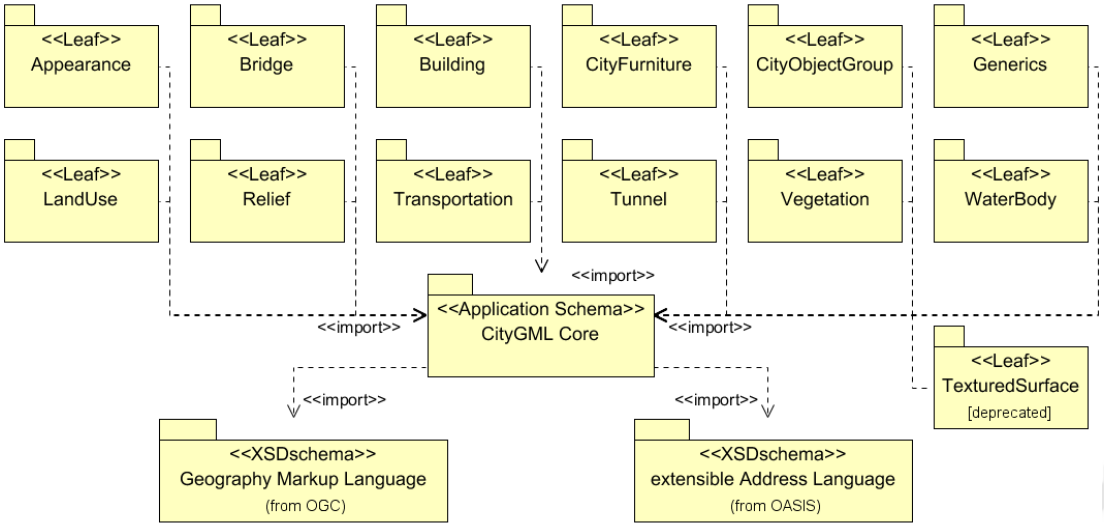

图 8. 展示CityGML独立模块及其模式依赖关系的UML包图。每个扩展模块（由叶子包表示）进一步导入GML 3.1.1模式定义，以表示其主题类的空间属性。为了便于阅读，省略了相应的依赖项。

表4: CityGML核心和主题扩展模块概述

模块名称	CityGML核心
XML命名空间标识符	http://www.opengis.net/citygml/2.0
XML模式文件	cityGMLBase.xsd
建议的命名空间前缀	core
模块说明	*CityGML core* 模块定义了CityGML数据模型的基本组件。主要包括抽象基类，从这些基类可以（传递）派生所有主题类。但是，在核心模块中定义了多个扩展模块常见的非抽象内容，例如基本数据类型。 核心模块导入了GML 3.1.1版本的XML模式定义文件和OASIS可扩展地址语言xAL

模块名称	外观
XML命名空间标识符	http://www.opengis.net/citygml/appearance/2.0
XML模式文件	appearance.xsd
建议的命名空间前缀	app
模块说明	*Appearance* 模块提供了对CityGML特征外观进行建模的方法，即特征表面的可观察属性。可以为每个城市对象存储外观数据。因此，在核心模块中定义的抽象基类 *CityObject_ ，基于CityGML的* Application Domain Extension__ 机制的附加属性进行扩充。因此， *Appearance* 模块直接影响所有主题扩展模块。

模块名称	桥梁
XML命名空间标识符	http://www.opengis.net/citygml/bridge/2.0

XML模式文件	bridge.xsd
建议的命名空间前缀	brid
模块说明	*Bridge* 模块允许对桥梁的主题和空间属性、桥梁部件、桥梁安装和内部桥梁结构的四个细节层次(LOD 1 - 4)进行表达。

模块名称	建筑
XML命名空间标识符	http://www.opengis.net/citygml/building/2.0
XML模式文件	building.xsd
建议的命名空间前缀	bldg
模块说明	*Building* 模块允许在五个细节层次（LOD 0-4）中表达建筑的主题和空间属性，建筑部件，建筑装置和室内建筑结构。

模块名称	城市家具
XML命名空间标识符	http://www.opengis.net/citygml/cityfurniture/2.0
XML模式文件	cityFurniture.xsd
建议的命名空间前缀	frn
模块说明	*CityFurniture* 模块用于表达城市中的城市家具对象。城市家具是不可移动的物体，如灯笼、交通标志、广告栏、长凳或公共汽车站，可以在交通区、居民区、广场或建筑区找到。

模块名称	城市对象组合
XML命名空间标识符	http://www.opengis.net/citygml/cityobjectgroup/2.0
XML模式文件	cityObjectGroup.xsd
建议的命名空间前缀	grp
模块说明	*CityObjectGroup* 模块为CityGML提供了一个分组概念。可以根据用户自定义的标准将任意城市对象聚合到组中，表达和传递城市模型的一部分。一个组可以根据特定的属性进一步分类。

模块名称	通用
XML命名空间标识符	http://www.opengis.net/citygml/generics/2.0
XML模式文件	generics.xsd
建议的命名空间前缀	gen

模块说明	*Generics* 模块提供了对CityGML数据模型的通用扩展，可用于建模和交换CityGML预定义的主题类没有涵盖的其他属性和特性。但是，只有当其他任何CityGML模块没有提供适当的主题类或属性时，才使用通用扩展。 为了表达通用属性， *Generics* 模块基于CityGML的 *Application Domain Extension* 机制的附加属性扩充了核心模块中定义的城市对象抽象基类 *_CityObject* 。因此， *Generics* 模块会对所有主题扩展模块产生影响。

模块名称	土地利用
XML命名空间标识符	http://www.opengis.net/citygml/landuse/2.0
XML模式文件	landUse.xsd
建议的命名空间前缀	luse
模块说明	*LandUse* 模块表达地表特定土地用途的区域。

模块名称	地形
XML命名空间标识符	http://www.opengis.net/citygml/relief/2.0
XML模式文件	relief.xsd
建议的命名空间前缀	dem
模块说明	*Relief* 模块在城市模型中表达地形。CityGML支持不同级别的地形表示，反映不同的精度或分辨率。地形可以是一个规则的栅格或网格，如TIN，断线，和离散多点。

模块名称	交通
XML命名空间标识符	http://www.opengis.net/citygml/transportation/2.0
XML模式文件	transportation.xsd
建议的命名空间前缀	tran
模块说明	*Transportation* 模块表达城市内的交通特征，如道路、轨道、铁路或广场。交通特征可以是线性网络或通过几何描述的三维表面。

模块名称	隧道
XML命名空间标识符	http://www.opengis.net/citygml/tunnel/2.0
XML模式文件	tunnel.xsd
建议的命名空间前缀	tun

模块说明	*Tunnel* 模块以四层细节（LOD 1-4）的形式表达隧道、隧道部件、隧道装置和内部隧道结构的主题和空间。

模块名称	植被
XML命名空间标识符	http://www.opengis.net/citygml/vegetation/2.0
XML模式文件	vegetation.xsd
建议的命名空间前缀	veg
模块说明	*Vegetation* 模块提供了主题类来表达植被对象。CityGML的植被模型分为单独的植被对象（如树木）和植被区域（如森林或其他植物群落）。

模块名称	水体
XML命名空间标识符	http://www.opengis.net/citygml/waterbody/2.0
XXML模式文件	waterbody.xsd
建议的命名空间前缀	wtr
模块说明	*WaterBody* 模块表达河流、运河、湖泊和盆地的主题和三维几何图形，但没有继承任何水文或其他动态信息。

模块名称	纹理表面[已弃用]
XML命名空间标识符	http://www.opengis.net/citygml/texturedsurface/2.0
XML模式文件	texturedSurface.xsd
建议的命名空间前缀	tex
模块说明	*TexturedSurface* 表面模块为3D表面提供视觉外观属性（颜色，光泽，透明度）和纹理。由于建模方法的固有限制，该模块被标记为已弃用，预计未来的CityGML版本中被删除。此模块提供的外观信息可以转换为CityGML的 *Appearance* 模块，不会丢失信息。因此， **强烈建议** 不要使用 *TexturedSurface* 模块。

7.2. CityGML专用文件（profile）

CityGML专用文件是主题扩展模块与CityGML核心模块之间的一个任意组合。每个CityGML实例文档应使用与所提供数据相适应的CityGML专用文件。一般来说，在实例文档中使用CityGML专用文件有两种方法:

1. **CityGML专用文件的定义内嵌在CityGML实例文档中**
   可以使用在XML模式实例命名空间 http://www.w3.org/2001/XMLSchema-instance （通常与前缀 *xsi* 相关联）中定义的 *schemaLocation* 属性，将CityGML专用文件绑定到实例文档。 *xsi:schemaLocation*

提供了一种定位实例文档中定义的命名空间的XML模式定义的方法。它的值是一个由空格分隔的统一资源标识符（URIs）列表，其中每一对包含一个命名空间，后面跟着该命名空间的XML模式定义的位置，通常是一个.xsd文件。 通过这种方式，各个CityGML模块的命名空间应在CityGML实例文档中定义。这个 *xsi:schemaLocation* 属性提供每个模块各自的XML模式定义的位置。附件G中给出的所有示例文件都遵循第一种方法。

2. **CityGML专用文件的定义由单独的XML模式定义文件提供**
   CityGML专用文件也可以由它自己的XML模式文件指定。这个模式文件应该通过导入相应的XML模式定义来组合适当的CityGML模块。为此，应使用XML模式命名空间中定义的 *import* 元素 http://www.w3.org/2001/XMLSchema （通常与前缀 *xs* 相关联）。对于 *xs:import* 元素，须声明导入的CityGML模块的命名空间以及命名空间的XML模式定义的位置。为了将CityGML专用文件应用到实例文档，须使用 *xsi:schemaLocation* 属性将专用文件的模式绑定到实例文档。CityGML专用文件的XML模式文件不得包含其它内容。 专用文件模式的 *targetNamespace* 应该与导入的CityGML模块的命名空间不同。与专用文件相关联的命名空间应该由实例文档的发起者确定，并且必须以未使用的全局唯一URI的形式给出。专用文件的XML模式文件必须对解析CityGML实例文档的所有人都可用（或在互联网上可访问）。

第二种方法由以下CityGML的基本专用文件的XML模式定义的示例来说明。由于基本专用文件是所有CityGML模块的并集，相应的XML模式定义将导入每个CityGML模块。通过这种方式，CityGML数据模型的所有组件都可以通过引用基本专用文件的实例文档进行交换。基本专用文件的模式定义随CityGML模式包一起提供，可以访问：http://schemas.opengis.net/citygml/profiles/base/2.0/CityGML.xsd 。

```
<xs:schema xmlns="http://www.opengis.net/citygml/profiles/base/2.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/citygml/profiles/base/2.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified" version="2.0.0">
<xs:import namespace="http://www.opengis.net/citygml/appearance/2.0"
schemaLocation="http://schemas.opengis.net/citygml/appearance/2.0/appearan
ce.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/bridge/2.0"
schemaLocation="http://schemas.opengis.net/citygml/bridge/2.0/bridge.xsd"/
>
<xs:import namespace="http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://schemas.opengis.net/citygml/building/2.0/building.x
sd"/>
<xs:import namespace="http://www.opengis.net/citygml/cityfurniture/2.0"
schemaLocation="http://schemas.opengis.net/citygml/cityfurniture/2.0/cityF
urniture.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/cityobjectgroup/2.0"
schemaLocation="http://schemas.opengis.net/citygml/cityobjectgroup/2.0/cit
yObjectGroup.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/generics/2.0"
schemaLocation="http://schemas.opengis.net/citygml/generics/2.0/generics.x
sd"/>
<xs:import namespace="http://www.opengis.net/citygml/landuse/2.0"
schemaLocation="http://schemas.opengis.net/citygml/landuse/2.0/landUse.xsd
"/>
<xs:import namespace="http://www.opengis.net/citygml/relief/2.0"
schemaLocation="http://schemas.opengis.net/citygml/relief/2.0/relief.xsd"/
>
<xs:import namespace="http://www.opengis.net/citygml/transportation/2.0"
schemaLocation="http://schemas.opengis.net/citygml/transportation/2.0/tran
sportation.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/tunnel/2.0"
schemaLocation="http://schemas.opengis.net/citygml/tunnel/2.0/tunnel.xsd"/
>
<xs:import namespace="http://www.opengis.net/citygml/vegetation/2.0"
schemaLocation="http://schemas.opengis.net/citygml/vegetation/2.0/vegetati
on.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/waterbody/2.0"
schemaLocation="http://schemas.opengis.net/citygml/waterbody/2.0/waterBody
.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/texturedsurface/2.0"
schemaLocation="http://schemas.opengis.net/citygml/texturedsurface/2.0/tex
turedSurface.xsd"/>
</xs:schema>
```

下面的CityGML数据集摘录说明了如何应用基本专用文件模式 *CityGML.xsd* 到CityGML实例文档。该数据集包含两个建筑对象和一个城市对象组合，由CityGML定义基本专用文件 *CityGML.xsd* 是使用根元素的

xsi:schemaLocation 属性引用的。因此，所有CityGML模块都由实例文档使用，不需要进一步引用CityGML模块的XML模式文档。

```
<core:CityModel xmlns="http://www.opengis.net/citygml/profiles/base/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0"
xmlns:grp="http://www.opengis.net/citygml/cityobjectgroup/2.0"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xAL="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/citygml/profiles/base/2.0
http://schemas.opengis.net/citygml/profiles/base/2.0/CityGML.xsd">
<core:cityObjectMember>
<bldg:Building gml:id="Build0815">
<core:externalReference>
<core:informationSystem>http://www.adv-online.de </core:informationSystem>
<core:externalObject>
<core:uri>urn:adv:oid:DEHE123400007001</core:uri>
</core:externalObject>
</core:externalReference>
<bldg:function
codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
uilding_function.xml">1000</bldg:function>
<bldg:yearOfConstruction>1985</bldg:yearOfConstruction>
<bldg:roofType
codeSpace="http://www.sig3d.org/codelists/standard//building/2.0/_Abstract
Building_roofType.xml">1030</bldg:roofType>
<bldg:measuredHeight uom="#m">8.0</bldg:measuredHeight>
<bldg:storeysAboveGround>2</bldg:storeysAboveGround>
<bldg:storeyHeightsAboveGround uom="#m">2.5
</bldg:storeyHeightsAboveGround>
<bldg:lod2Solid> ... </bldg:lod2Solid>
</bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
<bldg:Building gml:id="Build0817">
…
</bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
<grp:CityObjectGroup gml:id="Complex113">
<gml:name>Hotel complex 'Scenic View'</gml:name>
<grp:function>building group</grp:function>
<grp:groupMember role="main building" xlink:href="#Build0817"/>
<grp:groupMember xlink:href="#Build0815"/>
</grp:CityObjectGroup>
</core:cityObjectMember>
</core:CityModel>
```

8. 空间模型

CityGML要素的空间属性被用GML3几何模型中的对象来表示。该模型基于标准ISO 19107“空间模式”（Herring 2001），根据公认的边界表示法表示3D几何体（B-Rep，cf.Foley et al.1995）。CityGML实际上只使用GML3 几何包的一个子集，定义GML3的专用文件（profile）。该子集在图9和图10中描绘。此外，GML3的显式边界表示通过场景图概念进行了扩展，该概念允许隐式表示具有相同形状的要素的几何体，从而更有效地表示空间（第8.2章）。

8.1. 几何拓扑模型

GML3的几何模型由图元组成，图元可以组合成复合体、复合几何或聚合体。对于每个维度，都有一个几何单元：零维对象是一个*Point*，一个一维对象是一条*_Curve*，一个二维对象是一个*_Surface*,一个三维对象是一个*_Solid*（图9）。每个几何图形都有自己的坐标参考系统。一个实体由若干面构成，面由曲线构成。在CityGML中，一条曲线被定义为一条直线，因此只使用GML3的*LineString*类。CityGML中的表面由*Polygons*表达，它定义了一个平面几何，即边界和所有内部点都需要位于一个平面上。

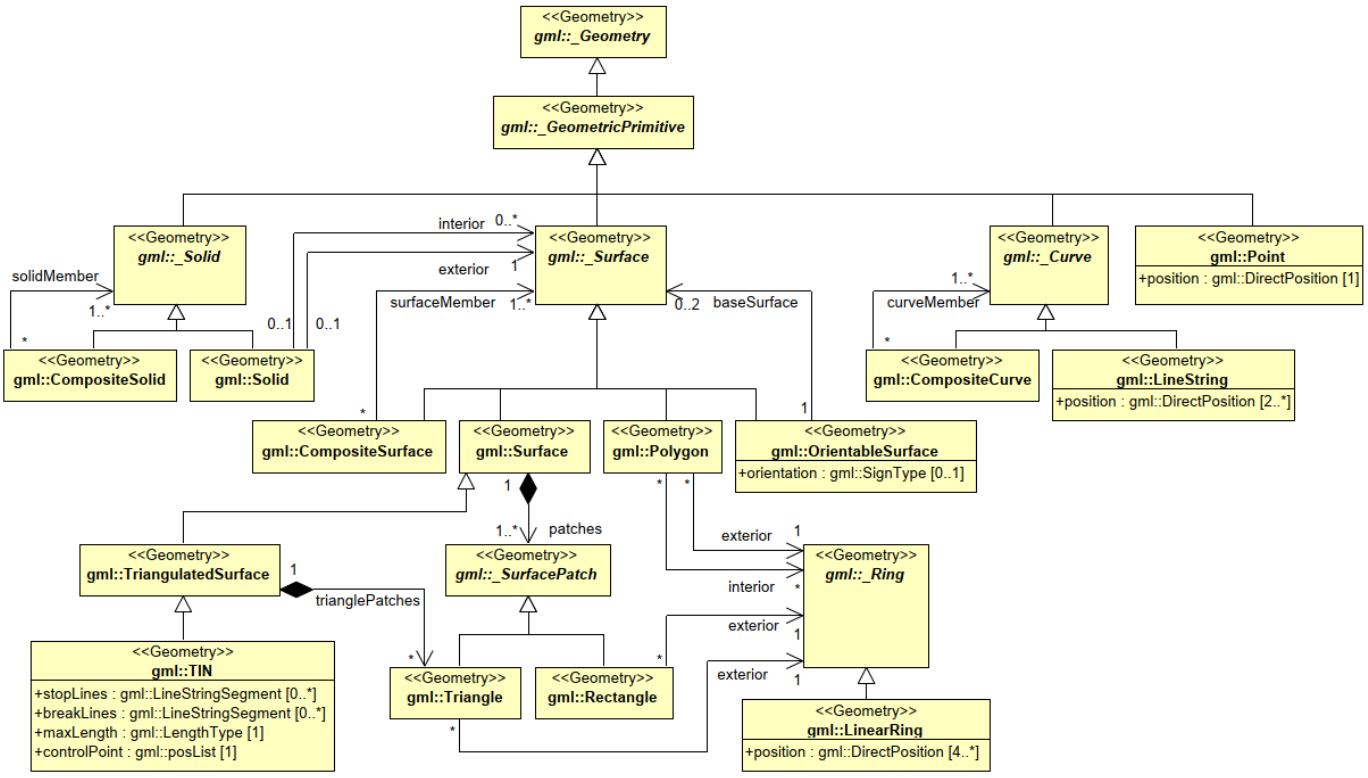

图 9. CityGML几何模型的UML图（GML3的子集和概要）：图元和组合。

组合的几何形状可以是图元的聚合体、复合物（见图11）。对于一个聚合体，组件之间的空间关系不受限制，它们可能是不相交的、重叠的、接触的或不连接的。GML3为每个维度：*MultiPoint*、*MultiCurve*、*MultiSurface*和*MultiSolid*，提供特殊聚合（见图10）。与聚合不同，复合体具有拓扑结构：其组件须不相交，不得重叠，最多允许其边界接触或共享部分边界。复合体是GML3提供的特殊复合物，只能包含相同维度的元素。它的元素也须是不相交的，但边界在拓扑上连接。复合物可以是一个*CompositeSolid*、*CompositeSurface*或*CompositeCurve*。（参见图9）。

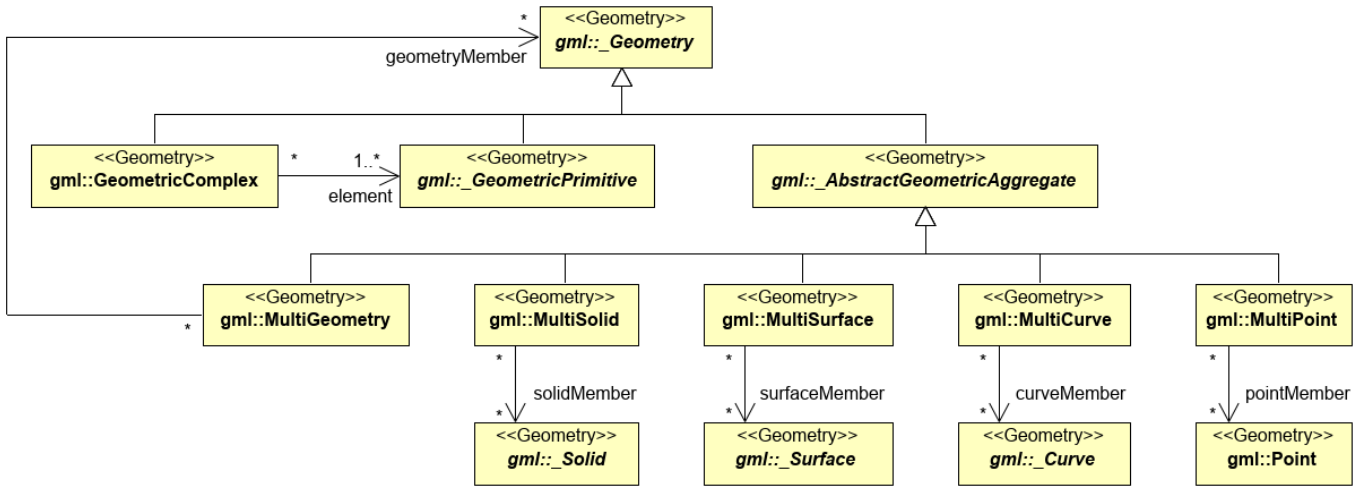

图 10. CityGML几何模型的UML图：复合体和聚合体

*OrientableSurface（定向面）*是具有明确方向，即可以区分前后两侧。这可用于将纹理指定给面的特定边，或在边界实体区分曲面的外部和内部。请注意，曲线和面在GML中有一个默认方向，该方向由定义点的顺序决定。因此，如果给定GML几何体的方向必须反转，则只需使用*OrientableSurface*。

*TriangulatedSurfaces（三角面）*是一种特殊的面，它是用来表示地形的不规则三角网。
当*TriangulatedSurfaces*是由显式*Triangles*组成时，子类*TIN*通过一组控制点以隐式方式表示三角剖分，定义三角形的节点。可以使用标准三角剖分方法（Delaunay三角剖分）重建三角剖分。此外，打断线和停止线定义了地形轮廓特征。

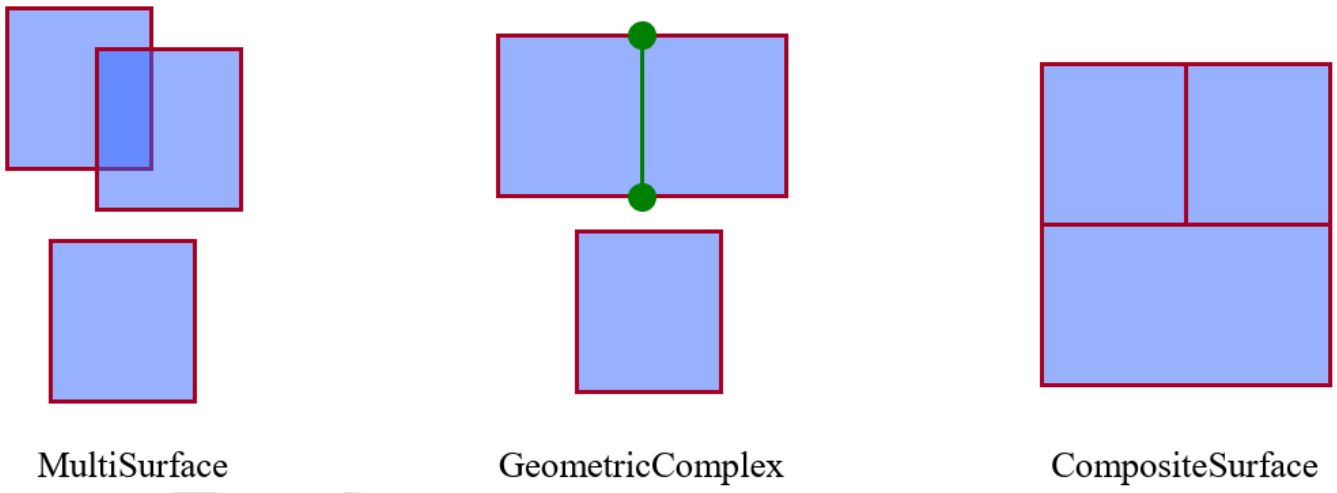

图 11. 组合几何图形

GML3复合模型实现了对应维度的每个单元类型的递归聚合模式。此聚合模式允许定义嵌套聚合（组件层次结构）。例如，建筑几何图形（*CompositeSolid*）可以由房屋几何图形（*CompositeSolid*）和车库几何图形（*Solid*）组成，而房屋几何图形进一步分解为屋顶几何图形（*Solid*）和房屋主体几何图形（*Solid*）。

CityGML提供了拓扑的显式建模，例如在要素或其他几何体之间共享几何体对象。空间的一部分仅由几何体对象表示一次，并由该几何体对象定义或限定的所有要素或更复杂的几何体引用，从而避免了冗余，保持了部件间的显式拓扑关系。基本上有三种情况。首先，两个要素在空间可以由相同的几何体上定义。例如，如果路径既是交通要素又是植被要素，则定义路径的面几何图形将同时由交通对象和植被对象引用。其次，几何图形可以在要素和其他几何单元之间共享。定义建筑墙的几何图元可以被引用两次：通过定义建筑几何图形的实体几何单元和墙

要素。第三，两个几何对象可能在边界上引用相同的几何图形。例如，一个建筑和一个相邻的车库可以用两个实体来表达，中间的接触区可以被两个实体引用，但只能表示一次。从图12可以看出，这需要划分各自的表面。通常，边界表示可见的表面。然而，使拓扑邻接显式，并允许删除一个组合对象的一部分，而不在剩余的聚合接触元素中留下洞。 虽然允许接触，但物体的渗透并不是为了避免同一空间的多重表达。但是，在CityGML中使用拓扑是可选的。

为了实现拓扑结构，CityGML使用了GML提供的*XLinks*XML概念。由不同的几何集合或不同的主题要素共享的每个几何对象都被分配了一个唯一的标识符，该标识符可以由使用*href*属性的GML几何属性引用。CityGML不部署内置的GML3拓扑包，后者提供了伴随几何图形的单独的拓扑对象。这种拓扑结构非常复杂精细。当数据集（可能包括或忽略拓扑）应该由相同的数据模型覆盖时，它缺乏灵活性。XLink拓扑简单灵活，几乎与显式的GML3拓扑模型一样强大。然而，XLink拓扑的一个缺点是拓扑连接的对象之间的导航只能单向执行（从聚合到其组件），而不像GML的内置拓扑那样（立即）双向执行。CityGML的拓扑表示示例在附录G.4中列出的数据集中给出。

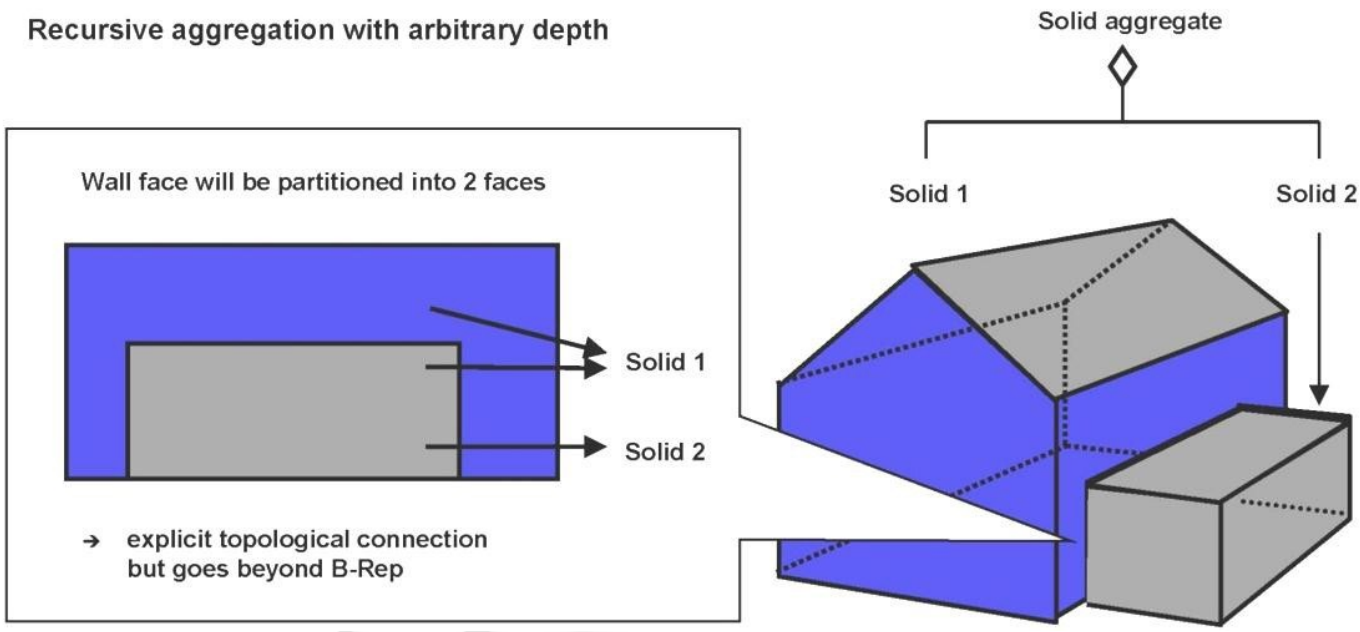

图 12. CityGML中对象和几何图形的递归聚合（图:IGG Uni Bonn）

下面摘录的CityGML示例文件定义了一个*gml:Polygon*，包含一个*gml:id wallSurface4711*，这是建筑的几何属性*lod2Solid*的一部分。与第一个建筑相邻的另一个建筑在其几何表示中引用了这个共享多边形。

```
<bldg:Building>
<bldg:lod2Solid>
<gml:surfaceMember>
<gml:Polygon gml:id="wallSurface4711">
<gml:exterior>
<gml:LinearRing>
<gml:pos srsDimension="3">32.0 31.0 2.5</gml:pos>
...
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
...
</bldg:lod2Solid>
</bldg:Building>
...
<bldg:Building>
<bldg:lod2Solid>
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#wallSurface4711"/>
</gml:OrientableSurface>
</gml:surfaceMember>
...
</bldg:lod2Solid>
</bldg:Building>
```

8.2. 空间参照系

当处理地理信息和虚拟三维城市模型时，精确的空间参考是至关重要的，也是将不同空间数据集集成到一个三维城市模型中的关键。CityGML继承了GML3处理坐标参考系统（CRS）的空间功能，CRS是GML 3.1.1中表示空间参考的常用方式。由于CityGML是一个真正的3D标准，几何元素与3D CRS相关联。只有少数情况CityGML允许引用二维几何元素（例如，在CityGML的*外观*模块中定义的*地理引用纹理*的*参考点*必须是二维的坐标值，参考章节9.4）。

通常，一个几何图形可以通过从抽象GML超类*GML:_Geometry*继承放入属性*srsName*指向其CRS定义。这可能是对权威组织如欧洲石油勘探集团（EPSG）提供的知名CRS定义的引用，也可能是指向在同一个CityGML实例文档中本地定义的CRS的指针。OGC文档“定义标识符URN在OGC命名空间”（cf. Whiteside 2009;OGC Doc. No.07-092r3）为CRS参考资料的URN编码提供最佳方案。其中，它描述了如何引用一个众所周知的3D CRS定义（如3D地理CRS），以及结合了两种或两种以上众所周知的CRS定义（例如，投影CRS用于平面参考，垂直CRS用于高度参考）的复合CRS。在附录G中给出了在CityGML实例文档中表示复合CRS的示例。

GML3还支持工程CRS的定义，这些CRS是在局部环境中使用的。例如，这可能是一个局部的三维笛卡尔坐标系，地球表面近似平地，因此忽略了地球曲率对要素几何的影响（参见GML 3.1.1规范文档的第12.1.4.4章）。局部工程CRS通常应用于AEC/FM领域，因此在将CAD数据或BIM模型集成到三维城市模型中时非常有用。附件 G.9 提供了一个示例，演示了在 CityGML 实例文档中定义工程 CRS 以及在要素几何中使用局部坐标值。工程CRS的定义需要一个锚点，它将本地坐标系的原点与地球表面上的一个点联系起来，以便于本地工程CRS的坐

标转换。

根据GML 3.1.1，如果一个几何元素上没有给出*srsName*属性，那么CRS应被指定为这个几何元素所处的更大上下文的一部分，例如一个几何集合。为了方便构造要素和要素集合实例，*gml:Envelope*（或*gml:Box*）上的*srsName*属性的值，也就是特征的*gml:boundedBy*属性的值，应被要素或集合成员的所有属性中所有直接表达的几何元素所继承，除非被本地存在的*srsName*推翻。因此，如果一个几何体使用了与其父要素的*gml:boundedBy*属性相同的CRS，它就没有必要携带*srsName*属性。CRS的继承一直延续到各级深度的嵌套，但是如果被本地的*srsName*所替代，那么新的CRS会被它的所有子代依次继承（参见GML 3.1.1规范文档的8.3章）。

强烈建议任何 CityGML 实例文档都要明确指定所有包含的几何元素的 CRS。如果实例文档要在外部与第三方交换或将与其他空间数据集整合，这一点尤为重要。在同一数据集内混合使用不同的CRS是可能的，并且符合GML 3.1.1的要求，而在*CityModel*特征集合上给出的单一CRS引用（参见第10.1章）简化了软件系统对数据集的处理。至于CityGML 2.0，这个建议是非规范性的，因此没有附带一致性类。这样做的主要原因是为了保持与CityGML 1.0的向后兼容性。

8.3. 隐式几何、原型对象、场景图概念

隐式几何的概念是对GML3几何模型的改进。例如，它被用于CityGML的建筑、桥梁、隧道和植被模型，以及城市家具和通用对象。隐式几何模型可以应用于CityGML不同主题领域的要素，以便在特定的细节层次模型（LOD）内对要素进行几何表示。因此，每个扩展模块可以定义空间属性，为其主题类提供隐式几何图形。隐式几何图形的概念是在CityGML核心模块中定义的（参见第10.1章）。隐式几何是CityGML空间模型的一部分，因此在这里进行描述。UML图如图13。相应的XML模式定义在附件A.1中提供。

隐式几何是一个几何对象，其形状只作为原型几何体存储一次，例如一棵树或其他植被对象，一个交通灯或交通标志。这个原型几何体对象被多次重复使用或引用，无论相应的要素出现在三维城市模型中的哪里。每一次出现都由一个与原型形状几何体的链接（在局部的笛卡尔坐标系中）、一个与原型的每个三维坐标相乘的变换矩阵和一个表示对象在世界坐标参考系中的基点的固定点来表达。这个参考点也定义了变换后坐标所属的CRS。为了确定隐式几何的绝对坐标，固定点坐标必须加到矩阵乘法结果中。变换矩阵负责了原型的旋转、缩放和局部平移。它是一个4x4矩阵，用齐次坐标即（x,y,z,1），与原型坐标相乘。这样，即使是投影也可以用变换矩阵来建模。

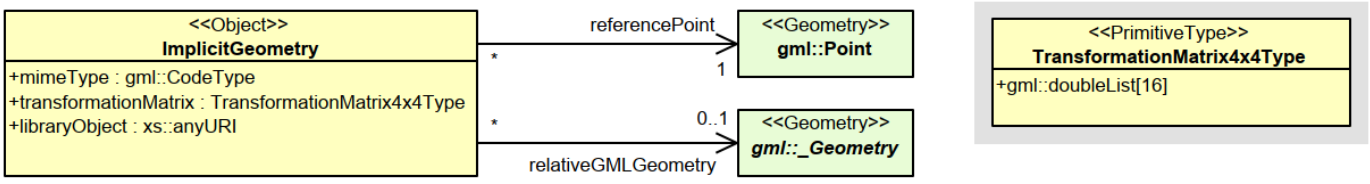

图 13. 隐式几何的 UML 图。前缀用于指示与模型元素相关的 XML 命名空间。没有前缀的元素名称是在CityGML核心模块中定义的。

在CityGML中使用隐式几何体概念可以提升空间效率。如：由于同种树木的形状可以被视为完全相同，因此如果要对大量树木中的每一棵树的详细几何形状进行显式建模，效率会很低。隐式几何图形的概念类似于计算机图形学领域用于表达*场景图*的*图元实例*概念（Foley等人，1995）。

implicit geometry，是指具有复杂形状的几何对象可以简单地用一个基点和一个变换来表示，从而在世界坐标系的某一特定位置展开对象形状。

*ImplicitGeometry*的形状可以用专有格式的外部文件来表示，例如VRML文件、DXF文件或3D Studio MAX文件。对隐式几何的引用可以通过指向本地或远程文件的URI来指定，甚至可以指向一个适当的网络服务。另外，

形状可以由GML3几何对象来定义。这样做的好处是，它可以在CityGML数据集中进行在线存储或交换。通常情况下，几何形状是在本地坐标系中定义的，原点位于对象的范围内或附近。如果形状是由URI引用的，还必须指定所表示对象的MIME类型（例如，VRML模型的 "model/vrml "或X3D模型的 "model/x3d+xml"）。

与使用绝对世界坐标表达物体的显式建模相比，三维物体几何体的隐式表达具有一些优势。它更节省空间，因此系统可以存储或处理更多的场景。由于3D图形卡支持场景图的概念，因此可以加快可视化的速度。此外，由于只需交换库中的对象，因此便于使用不同形状版本的对象，例如不同季节的对象（见图65的例子）。

XML命名空间

*CityGML core*模块的XML命名空间定义了隐含几何图形的概念，由统一资源标识符（URI）http://www.opengis.net/citygml/2.0 标识。在核心模块的XML Schema定义中，这个URI也被用来标识默认的命名空间。

ImplicitGeometryType, ImplicitRepresentationPropertyType

```
<xs:complexType name="ImplicitGeometryType">
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType">
<xs:sequence>
<xs:element name="mimeType" type="gml:CodeType" minOccurs="0"/>
<xs:element name="transformationMatrix" type="TransformationMatrix4x4Type"
minOccurs="0"/>
<xs:element name="libraryObject" type="xs:anyURI" minOccurs="0"/>
<xs:element name="relativeGMLGeometry" type="gml:GeometryPropertyType"
minOccurs="0"/>
<xs:element name="referencePoint" type="gml:PointPropertyType"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="ImplicitGeometry" type="ImplicitGeometryType"
substitutionGroup="gml:_GML"/>
<!--
================================================================================
================ -->
<xs:complexType name="ImplicitRepresentationPropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="ImplicitGeometry"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

8.3.1. 代码列表

*ImplicitGeometry*的*mimeType*属性被指定为*gml:CodeType*。这个属性的值可以在一个代码列表中列举出来。关于这一代码表的建议见附件C.6。

8.3.2. CityGML数据集示例

下面的城市家具对象（参见第10.9章）是一个隐式几何的例子，它由LOD2中的几何对象进行表达。

```
<frn:CityFurniture>
<!-- class “traffic”; as specified in the code list proposed by the SIG 3D
(cf. annex C.4) -->
<frn:class codeSpace=
"http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_c
lass.xml">1000</frn:class>
<!-- function “traffic light”; as specified in the code list proposed by
the SIG 3D (cf. annex C.4) -->
<frn:function codeSpace=
"http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_f
unction.xml">1080</frn:function>
<frn:lod2ImplicitRepresentation>
<core:ImplicitGeometry>
<core:mimeType>model/vrml</core:mimeType>
<core:libraryObject>
http://www.some-3d-library.com/3D/furnitures/TrafficLight434.wrl
</core:libraryObject>
<core:referencePoint>
<gml:Point srsName="
urn:ogc:def:crs,crs:EPSG:6.12:31467,crs:EPSG:6.12:5783">
<gml:pos srsDimension="3">5793898.77 3603845.54 44.8</gml:pos>
</gml:Point>
</core:referencePoint>
</core:ImplicitGeometry>
</frn:lod2ImplicitRepresentation>
</frn:CityFurniture>
```

交通信号灯的几何形状（根据附件C.4中提出的代码表，等级为 "1000"、功能为 "1080 "的城市家具）由一个VRML文件定义，该文件由一个URL指定。这个库中的对象是在本地坐标系中定义的，通过添加参考点的坐标将其转换为实际位置。

下面是一个CityGML文件的片段，提供了一个更复杂的隐式几何的例子。

```
<frn:CityFurniture>
<!-- class “traffic”; as specified in the code list proposed by the SIG 3D
(cf. annex C.4) -->
<frn:class>1000</frn:class>
<!-- function “traffic light”; as specified in the code list proposed by
the SIG 3D (cf. annex C.4) -->
<frn:function>1080</frn:function>
<frn:lod2ImplicitRepresentation>
<core:ImplicitGeometry>
<core:mimeType>model/vrml</core:mimeType>
<core:transformationMatrix>
0.866025 -0.5 0 0.7
0.5 0.866025 0 0.8
0 0 1 0
0 0 0 1
</core:transformationMatrix>
<core:libraryObject>
http://www.some-3d-library.com/3D/furnitures/TrafficLight434.wrl
</core:libraryObject>
<core:referencePoint>
<gml:Point srsName="
urn:ogc:def:crs,crs:EPSG:6.12:31467,crs:EPSG:6.12:5783">
<gml:pos srsDimension="3">5793898.77 3603845.54 44.8</gml:pos>
</gml:Point>
</core:referencePoint>
</core:ImplicitGeometry>
</frn:lod2ImplicitRepresentation>
</frn:CityFurniture>
```

除第一个例子外，还指定了一个变换矩阵。它是一个齐次矩阵，以行方式进行序列化，即列表中的前四个条目表示矩阵的第一行。矩阵结合了矢量的平移（0.7,0.8,0）—本地参考系的原点不是物体的中心，绕z轴旋转30度（*cos(30)=0.866025*和*sin(30)=0.5*）。为使交通信号灯与道路保持一致，这个旋转是必要的。交通信号灯的实际位置计算如下：

1. 将VRML文件中的每个点（齐次坐标）乘以变换矩阵。
2. 对每一个结果点，加上参考点(5793898.77,3603845.54,44.8,1)^{T}，得出城市家具的实际几何形状。

8.3.3. 一致性要求

基本要求

1. 了使用隐式几何概念对某一要素进行几何表达，该要素的相应主题类应定义一个隐式表示的空间属性*ImplicitRepresentationPropertyType*。因此，对于所有CityGML扩展模块，只能将类型*ImplicitRepresentationPropertyType*用于提供隐式几何的空间属性。
2. 如果隐式几何的形状是由URI隐式几何体的*libraryObject*属性（类型：*xs:anyURI*）引用的，也必须指定被表示对象的MIME类型。

参考文献的完整性

*ImplicitRepresentationPropertyType*类型可以包含一个内联的隐式几何元素，或者使用 GML 3.1.1 的 XLink 概念对远程隐式几何元素进行 XLink 引用。在 后一种情况下，*ImplicitRepresentationPropertyType*的相应属性的*xlink:href*属性只能指向远程隐式几何元素（其中远程隐式几何元素位于另一文档或同一文档的其他地方）。必须提供包含的元素或引用，但不能同时提供。

9. 外观模型

除了空间属性之外，CityGML 的特征还有外观—要素表面的可观察属性。外观并不局限于视觉数据，而且可以表达被称为主题的任意类别，如红外辐射、噪声污染或地震引起的结构应力。每个LOD可以为一个特定的主题定义一个单独的表象。一个外观由每个表面几何对象的数据组成即表面数据。一个表面几何对象可以有多个主题的表面数据。同样，表面数据也可以由多个表面几何对象共享（如道路铺装）。最后，表面数据值既可以在整个表面上是恒定的，也可以随表面内的确切位置移动而变化。 CityGML 的外观模型是在扩展模块*外观*中定义的（参见第 7 章）。外观模型UML图如图14所示，XML 模式定义见附件A.2。

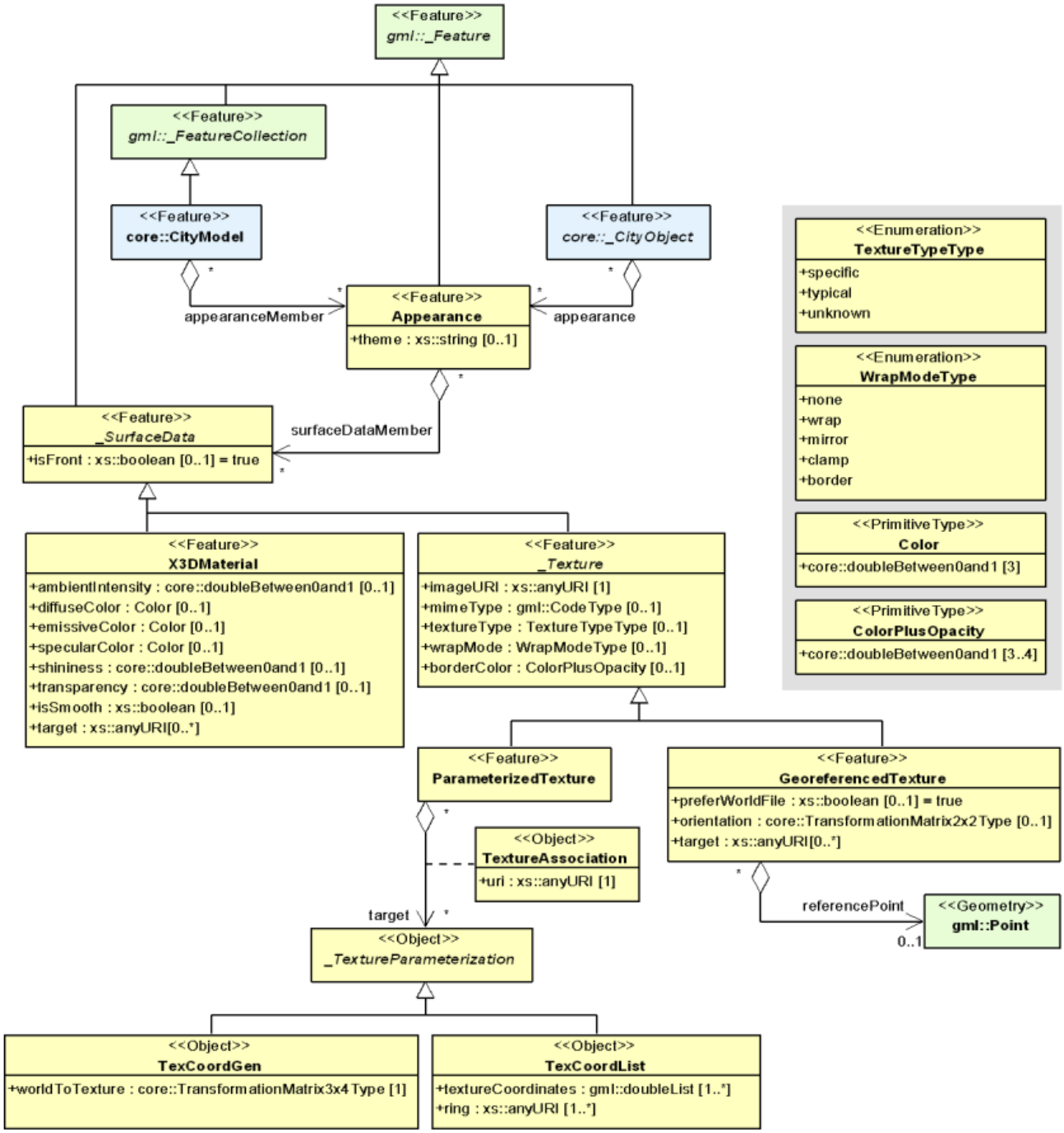

图 14. CityGML的外观模型的UML图。前缀用来表示与模型元素相关联的 XML 命名空间，没有前缀的元素名称是在 CityGML 外观模块中定义的。

在CityGML的外观模型中，主题只用一个标识符来表达。一个城市模型的外观对于一个给定的主题来说，是由一组引用这个主题的外观对象来定义的。因此，属于同一主题的外观对象组成了一个虚拟组。它们可以被包含在CityGML数据集的不同位置。此外，一个CityGML数据集可能包含多个主题。一个外观对象收集了与特定主题相关的表面数据，可以是单个要素，也可以是任何LOD中的整个城市模型。表面数据由类*_SurfaceData*及其子类的对象来表达，每个对象都涵盖了表面几何对象的整个区域。表面数据和表面几何对象之间的关系由*URI（统一资源标识符）*链接表示，从一个*_SurfaceData* 对象到一个类型为 *gml: AbstractSurfaceType*或类型*gml: MultiSurface*。

固定的表面属性被建模为材质，随表面位置变化的表面属性被建模为纹理。每个表面几何对象可以在每个主题和侧面同时拥有一个材质和一个纹理。这允许同时提供一个恒定的近似值和一个复杂的表面属性测量值。一个应用程序负责为其任务（如分析或渲染）选择合适的属性表示。具体的混合没有定义，因为这超出了CityGML的范围。如果一个表面几何对象要接收多个纹理或材质，每个纹理或材质都需要一个单独的主题。主题的混合或其用法在 CityGML 中没有定义，由应用程序自行决定。

XML命名空间

CityGML外观模块的XML命名空间由统一资源标识符（URI） http://www.opengis.net/citygml/appearance/2.0 定义。在*外观*模块的XML 模式定义中，这个URI也被用来标识默认的命名空间。

9.1. 外观、要素和几何图形之间的关系

尽管表面数据和表面之间有密切的关系，但表面数据是单独存储在要素中的，以保留原始的GML几何模型。表面数据不是各自目标表面几何对象的属性，而是每个表面数据对象保持一组URI，指定目标面几何对象的*gml:ids*（类型为*gml: AbstractSurfaceType*或*gml: MultiSurface*）。在复合或集合目标表面的情况下，表面数据对象被分配给所有包含的表面。其他目标类型如要素、实体或*gml: AbstractSurfacePatchType*（包括*gml: Triangle*）是无效的，尽管XML模式语言不能正式表达对URI目标类型的约束。对于表面数据值到表面补丁的确切映射功能，请参考各自的表面数据类型描述。

有效的目标类型限制为*gml: AbstractSurfaceType*和*gml: MultiSurface*，不包括基于GML几何模型和在CityGML中使用的*gml: AbstractSurfacePatchType*。一般来说，GML的表面是用*gml: AbstractSurfaceType*的子类来表示。这样的表面要求是连续的。一个*gml: MultiSurface*不需要满足这个要求，因此它不是*gml: AbstractSurfaceType*（参见8.1）。由于获得的真实世界的表面往往不能保证是连续的，CityGML允许*gml: MultiSurface*在不同的地方表示一个要素的边界，作为连续表面的替代。为了类似于*gml: CompositeSurface*来处理这些表面，表面数据对象被允许链接到*gml: MultiSurface*对象。因此，一个*gml: AbstractGMLType*（它包括*gml: Triangle*和*gml: Rectangle*）不能接收一个*gml:id*，也不能被引用。

每个表面几何对象的每个主题最多可以有一个有效的正面材料、一个有效的背面材料、一个有效的正面纹理和一个有效的背面纹理。如果多个表面数据对象被分配到一个表面几何对象上，会选择一个激活。由于嵌套的表面定义而产生的多个间接分配是通过覆盖来解决的，例如，一个*gml:Polygon*的正面材质是通过覆盖其父代*gml:CompositeSurface*的正面材质而激活。多重直接分配即一个表面几何对象的*gml:id*在一个主题中，不允许被多次引用，并通过在冲突的表面数据对象中选择一个来解决实现依赖的问题。因此，需要避免在一个主题中进行多次直接分配。

每个*_CityObject*要素都可以存储表面数据。因此，表面数据排列在CityGML数据集的要素层次中。然后，表面数据使用URI链接到其目标面。尽管这个链接机制允许在要素层次结构中链接到另一个要素表面，建议仍遵循局部性的原则。表面数据的存储方式应使链接的表面只属于包含的城市对象要素及其子对象。“全局性的”表面数据应该与城市模型一起存储。遵循定位原则还可以确保从WFS中检索的CityObjects包含相应的外观信息。

定位原则允许下面的算法找到所有相关的*_SurfaceData*对象，在一个给定的*_CityObject*中引用表面几何对象（类型为*gml:AbstractSurfaceType*或*gml:MultiSurface*）。

清单1：找到所有相关的_SurfaceData对象的算法，这些对象是在一个给定的_CityObject中指定表面几何对象（类型为gml:AbstractSurfaceType或gml:MultiSurface）。

```
*function findSurfaceData* ::
__in__:  gmlSurface, cityObject +
__out__:  frontMaterial, frontTexture, backMaterial, backTexture

1: frontMaterial := empty
2: frontTexture := empty
3: backMaterial := empty
4: backTexture := empty
5: flip := false
6:
7: while (gmlSurface) { // traverse the geometry hierarchy from inner to
outer
8: cObj := cityObject // start from the innermost cityobject
9:
10: while (cObj) { // traverse the cityobject hierarchy for the current
geometry object
11: // search all surfaceData objects in all appearance containers
12: foreach (appearance in cObj) {
13: foreach (surfaceData in appearance) {
14: if (surfaceData refers to gmlSurface) { // if a surfaceData object
refers to the geometry object, check its category
15: if (flip) { // consider flipping
16: // only pick the first surfaceData for a particular category
17: if (surfaceData is frontside material AND backMaterial is empty) {
18: backMaterial := surfaceData
19: }
20: if (surfaceData is frontside texture AND backTexture is empty) {
21: backTexture := surfaceData
22: }
23: if (surfaceData is backside material AND frontMaterial is empty) {
24: frontMaterial := surfaceData
25: }
26: if (surfaceData is backside texture AND frontTexture is empty) {
27: frontTexture := surfaceData
28: }
29: } else {
30: // only pick the first surfaceData for a particular category
31: if (surfaceData is frontside material AND frontMaterial is empty) {
32: frontMaterial := surfaceData
33: }
34: if (surfaceData is frontside texture AND frontTexture is empty) {
35: frontTexture := surfaceData
36: }
```

```
37: if (surfaceData is backside material AND backMaterial is empty) {
38: backMaterial := surfaceData
39: }
40: if (surfaceData is backside texture AND backTexture is empty) {
41: backTexture := surfaceData
42: }
43: }
44:
45: // shortcut: could stop here if all 4 categories have been found
46: }
47: }
48: }
49: cObj := cObj.parent // this also includes the global CityModel
50: }
51: gmlSurface := gmlSurface.parent // this also includes a root
gml:MultiSurface
52: if (gmlSurface isA gml:OrientableSurface AND gmlSurface.orientation is
negative) {
53: negate flip
54: }
55: }
```

对*_SurfaceData*对象的*isFront*属性的评估需要考虑到*gml: OrientableSurfaces*，因为它们可以翻转一个表面的方向。假设一个*gml: OrientableSurfaces os*，它翻转了它的基本表面*bs*。一个以*bs*为目标的正面纹理*t*将出现在*bs*的实际正面上。如果*t*的目标是*os*，它将出现在*bs*的背面。如果*t*同时以*os*和*bs*为目标，它就会出现在*bs*的两边，因为它成为了正面和背面的纹理。

XLinks影响了伪代码中的层次结构遍历。一般来说，表面数据和几何对象的分离需要重新评估表面数据的分配，因为每次出现几何对象的时候，都需要在相应的*_CityObject*的背景中进行评估。在算法中逐步提高（几何体或*_CityObject*）的层次结构，即为了这个算法的目的，被引用的对象在概念上被复制到所指的XLink的位置。特别是，这适用于隐式几何对象。如果一个隐式几何对象包含GML几何（在*relativeGMLGeometry*属性中），表面数据的分配就需要在每个引用的*_CityObject*的背景中被重新评估。因此，一个给定的*隐式几何*对象的外观（非相对几何图形）可能会在其出现的时候有所不同。如果所有需要的表面数据对象被放置在__外观_对象中，并且后者被储存在外观对象中，则会产生一致的外观结果：

1. 在包含原始*隐式几何*的*_CityObject*中，用XLinks引用所有*隐式几何*中引用的*_CityObject*中的相同*外观*对象，或
2. 在全局的*CityModel*中。

9.2. 外观和表面数据

特征类*Appearance*定义了一个表面数据对象的容器。它提供了所有包含的表面数据对象都与之相关的*theme*。一个CityGML文件中所有具有相同主题的外观对象被认为是一个组。表面数据对象被存储在*surfaceDataMember*属性中。它们可以作为远程属性同时在多个主题中使用。 特征类*__SurfaceData*是材质和纹理的基类。它唯一的元素是布尔标志*isFront*，它决定了一个表面数据对象适用于哪一表面。请注意，外观模型的所有类都支持CityGML的ADE机制（参见第6.12和10.13章）。应用特定扩展的钩子是由元素"*_GenericApplicationPropertyOf* … "实现的。

AppearanceType, Appearance, AppearancePropertyType

```
<xs:complexType name="AppearanceType">
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="theme" type="xs:string" minOccurs="0"/>
<xs:element name="surfaceDataMember" type="SurfaceDataPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfAppearance" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="Appearance" type="AppearanceType" substitutionGroup=
"gml:_Feature"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAppearance" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="AppearancePropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="Appearance"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

appearanceMember, appearance

```
<xs:element name="appearanceMember" type="gml:FeaturePropertyType"
substitutionGroup="gml:featureMember"/>
<!--
================================================================================
===============-->
<xs:elementname="appearance"type="AppearancePropertyType"substitutionGroup
="core:_GenericApplicationPropertyOfCityObject"/>
```

*appearanceMember*的定义允许在一个城市模型特征集合中任意排列*_CityObject* 特征和*Appearance*特征（

参见第10.1章）。

为了在一个单一的*_CityObject*特征中存储外观信息，核心模块的相应抽象类*_CityObject*是由*外观*属性元素来扩展的。这个额外的*外观*属性是使用CityGML的*应用领域扩展*机制扩展到*_CityObject*中的（参见第10.13章）。通过这种方式，*_CityObject*的每个主题子类都继承了这个属性。因此，*外观*模块对每个定义*_CityObject* 的主题子类的扩展模块都有影响。

AbstractSurfaceDataType, __SurfaceData, SurfaceDataPropertyType

```
<xs:complexType name="AbstractSurfaceDataType" abstract="true">
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="isFront" type="xs:boolean" default="true" minOccurs="0
"/>
<xs:element ref="_GenericApplicationPropertyOfSurfaceData" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_SurfaceData" type="AbstractSurfaceDataType" abstract=
"true" substitutionGroup="gml:_Feature"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfSurfaceData" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="SurfaceDataPropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="_SurfaceData" minOccurs="0"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

9.3. 材质

材质定义的光反射属性对整个面几何对象来说是不变的。*X3DMaterial*类的定义来自X3D和COLLADA规范（参考X3D、COLLADA规范）。*diffuseColor*定义了漫反射光的颜色。 *specularColor*定义了镜面反射的颜色。*emissiveColor*是表面自身的颜色。所有的颜色都使用RGB值，红、绿、蓝的取值在0和1之间。透明度是使用*transparency*元素单独定义的，其中0代表完全不透明，1代表完全透明。*ambientIntensity*定义了在不考虑

光源的情况下，漫反射颜色的最小百分比。*shininess*控制镜面高光的锐度。0产生柔和的光亮，而1则产生鲜明的高光。*isSmooth*给出了法线插值的提示。如果这个布尔标志被设置为 "true"，顶点法线应该被用于着色（Gouraud着色）。否则，法线在面补丁是不变的（平面着色）。

目标表面是由目标元素指定的。每个元素包含一个目标面几何对象的URI（类型为*gml: AbstractSurfaceType*或*gml: MultiSurface*）。

X3DMaterialType, X3DMaterial

```
<xs:complexType name="X3DMaterialType">
<xs:complexContent>
<xs:extension base="AbstractSurfaceDataType">
<xs:sequence>
<xs:element name="ambientIntensity" type="core:doubleBetween0and1"
default="0.2" minOccurs="0"/>
<xs:element name="diffuseColor" type="Color" default="0.8 0.8 0.8"
minOccurs="0"/>
<xs:element name="emissiveColor" type="Color" default="0.0 0.0 0.0"
minOccurs="0"/>
<xs:element name="specularColor" type="Color" default="1.0 1.0 1.0"
minOccurs="0"/>
<xs:element name="shininess" type="core:doubleBetween0and1" default="0.2"
minOccurs="0"/>
<xs:element name="transparency" type="core:doubleBetween0and1" default=
"0.0" minOccurs="0"/>
<xs:element name="isSmooth" type="xs:boolean" default="false" minOccurs="
0"/>
<xs:element name="target" type="xs:anyURI" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfX3DMaterial" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
============== -->
<xs:element name="X3DMaterial" type="X3DMaterialType" substitutionGroup=
"_SurfaceData"/>
<!--
============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfX3DMaterial" type=
"xs:anyType" abstract="true"/>
```

9.4. 纹理和纹理映射

纹理的抽象基类是*_Texture*。CityGML中的纹理总是基于栅格的2D纹理。栅格图像由*imageURI*使用URI指定，可以是一个任意的图像数据资源，甚至是一个预先格式化的网络服务的请求。图像数据格式可以使用*mimeType*元素中的标准MIME类型来定义。

纹理可以由*textureType*属性来限定。*textureType*区分了特定对象的（*特定*）纹理和该对象表面的典型（*典型*）原型纹理。纹理也可以被分类为*unknown*。

纹理包装的规范是采用COLLADA标准。当访问底层图像栅格外部的纹理时，需要进行纹理包装。 *wrapMode*可取以下五个值之一（图15说明了这些包装模式的效果）。

1. *none* - 产生的颜色是完全透明的
2. *wrap* - 纹理被重复
3. *mirror* - 纹理被重复并镜像化
4. *clamp* - 纹理被夹在其边缘上
5. *border* - 产生的颜色由*borderColor*元素指定（RGBA）

在包装模式的*mirror*中，纹理图像在水平和垂直方向上都是重复的，类似于包装模式的*wrap*填补纹理空间。与*wrap*不同的是，每次重复的结果都是沿着重复的方向翻转前一个纹理部分。这种做法消除了包装纹理的边缘的约束，并形成一个无缝的纹理。

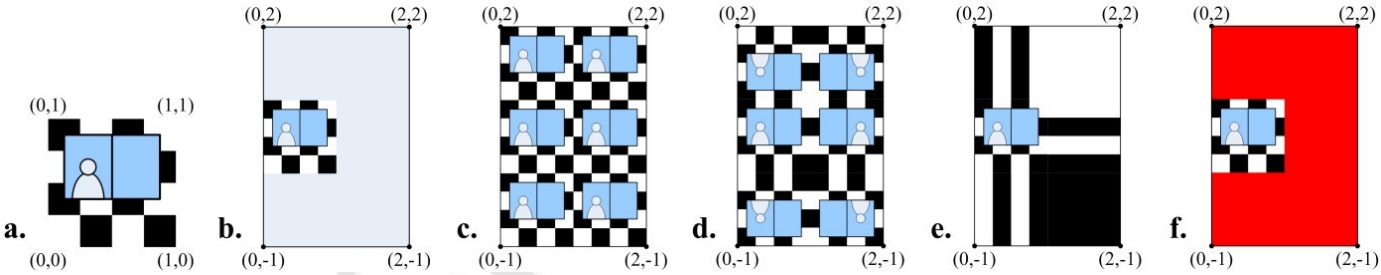

图 15. 一个纹理（a）使用不同的包装模式应用于一个外墙。（b）无，（c）包装，（d）镜像，（e）管夹和（f）边界。边界颜色为红色。数字表示纹理坐标（图片：Hasso-Plattner-Institut）。

AbstractTextureType, __Texture, WrapModeType, TextureTypeType

```
<xs:complexType name="AbstractTextureType" abstract="true">
<xs:complexContent>
<xs:extension base="AbstractSurfaceDataType">
<xs:sequence>
<xs:element name="imageURI" type="xs:anyURI"/>
<xs:element name="mimeType" type="gml:CodeType" minOccurs="0"/>
<xs:element name="textureType" type="TextureTypeType" minOccurs="0"/>
<xs:element name="wrapMode" type="WrapModeType" minOccurs="0"/>
<xs:element name="borderColor" type="ColorPlusOpacity" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfTexture" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_Texture" type="AbstractTextureType" abstract="true"
substitutionGroup="_SurfaceData"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTexture" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:simpleType name="WrapModeType">
<xs:restriction base="xs:string">
<xs:enumeration value="none"/>
<xs:enumeration value="wrap"/>
<xs:enumeration value="mirror"/>
<xs:enumeration value="clamp"/>
<xs:enumeration value="border"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
============== -->
<xs:simpleType name="TextureTypeType">
<xs:restriction base="xs:string">
<xs:enumeration value="specific"/>
<xs:enumeration value="typical"/>
<xs:enumeration value="unknown"/>
</xs:restriction>
</xs:simpleType>
```

根据纹理参数化，即从表面上的一个位置到纹理图像中的一个位置的映射函数，*_Texture*被进一步专门化。CityGML使用纹理空间的概念，纹理图像总是占据$[0,1]^2$区域，而不考虑实际的图像大小或长宽比。图像的左下角位于原点（有些图形API可能使用其他约定，需要进行纹理坐标转换）。必须知道每个面几何对象的映射函数，才能接受纹理。

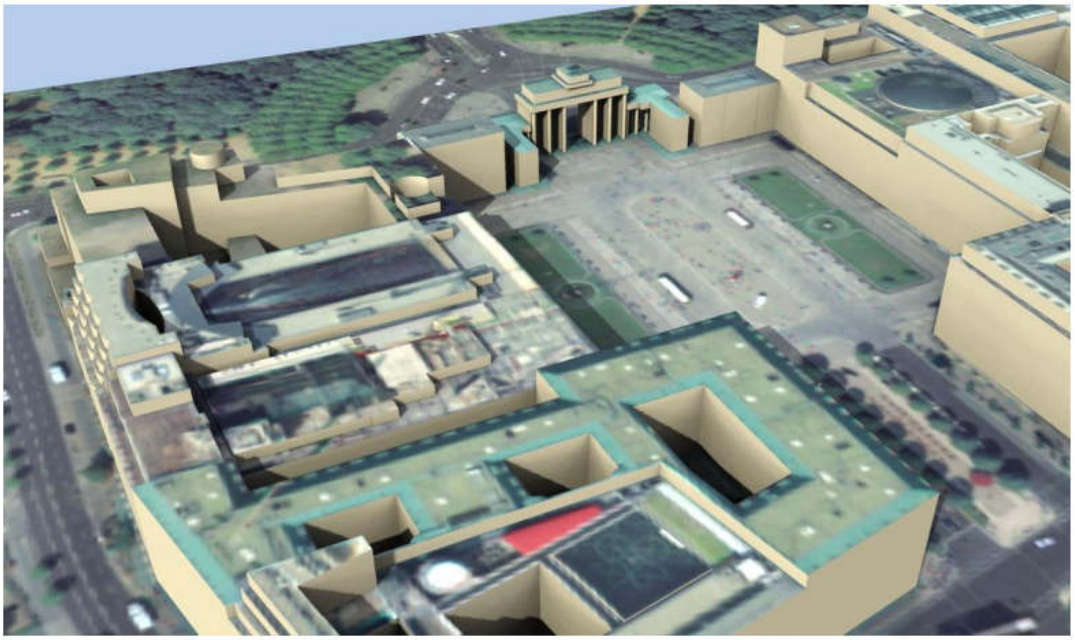

图 16. 应用于地面和屋顶的地理参考纹理（来源：柏林参议院，Hasso-Plattner-Institut）。

*GeoreferencedTexture*类描述了一个使用平面投影的纹理。因此，使用*GeoreferencedTexture*对垂直表面设置纹理是没有意义的。这样的纹理具有独特的映射功能，通常与图像文件一起提供（如地理参考的TIFF）或作为单独的ESRI世界文件。外部地理参考的搜索顺序是由布尔标志*preferWorldFile*决定的。如果这个标志被设置为"true"（其默认值），首先会寻找世界文件，只有在没有找到的情况下才会使用图像数据的地理参考。如果*preferWorldFile*为“false”，那么只有在没有图像数据的地理参考的情况下才会使用世界文件。

另外，CityGML允许内部指定一个类似于世界文件的地理参考。这个内部地理参考规范总是优先于任何外部地理参考。 *referencePoint*定义了世界空间中左上角图像像素的中心位置，对应于ESRI世界文件中的值5和6。由于*GeoreferencedTexture*使用的是平面投影，所以*referencePoint*是二维的。 *orientation*以2x2矩阵的形式定义了图像的旋转和缩放（4个双精度数字的列表，以行为主的顺序，对应于ESRI世界文件中的值1、3、2和4）。这种转换的CRS与*referencePoint*的CRS相同。该CRS中的一个平面点$(x,y)^T$通过以下公式被转换为纹理空间中的一个点$(s,t)^T$：

$$\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} 1/w & 0 \\ 0 & -1/h \end{pmatrix} \cdot M^{-1} \cdot \left(\begin{pmatrix} x \\ y \end{pmatrix} - P_R \right) + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

*M*表示*orientation*，P_R表示*referencePoint*，*w*表示图像的宽度（像素），*h*表示图像的高度（像素）。这种变换补偿了ESRI世界文件中使用的图像坐标系统（原点在左上角，X轴向右，Y轴向下）和CityGML中的纹理空间（原点在左下角，X轴向右，Y轴向上）之间的差异。

如果既没有给出内部地理参考，也没有给出外部地理参考，那么*GeoreferencedTexture*是无效的。每个目标表面几何对象是由一个*target*元素中的URI指定的。所有目标表面几何对象都共享地理参考所定义的映射功能。不允许有其他映射功能。请注意，从*gml: AbstractFeatureType* 继承的*gml: boundedBy*属性可以被设置为有效图像数据的边界框，以进行空间查询。图16显示了应用于地面和所有屋顶表面的地理参考纹理。

GeoreferencedTextureType, GeoreferencedTexture

```
<xs:complexType name="GeoreferencedTextureType">
<xs:complexContent>
<xs:extension base="AbstractTextureType">
<xs:sequence>
<xs:element name="preferWorldFile" type="xs:boolean" default="true"
minOccurs="0"/>
<xs:element name="referencePoint" type="gml:PointPropertyType" minOccurs=
"0"/>
<xs:element name="orientation" type="core:TransformationMatrix2x2Type"
minOccurs="0"/>
<xs:element name="target" type="xs:anyURI" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfGeoreferencedTexture"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="GeoreferencedTexture" type="GeoreferencedTextureType"
substitutionGroup="_Texture"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfGeoreferencedTexture" type
="xs:anyType" abstract="true"/>
```

*ParameterizedTexture*类描述了一个具有目标依赖映射功能的纹理。映射是由*_TextureParameterization*类的子类定义的，作为链接到目标面几何对象的一个属性。每个目标面几何对象都在一个单独的*target*元素的*uri*属性中被指定为URI。因为*target*实现了*gml: AssociationAttributeGroup*，它允许引用一个远程的*_TextureParameterization*对象（使用*xlink:href*属性），比如用于在不同主题的目标或纹理之间共享一个映射功能。映射函数既可以使用纹理坐标的概念（通过*TexCoordList*类），也可以使用从世界空间到纹理空间的变换矩阵（通过*TexCoordGen*类）。

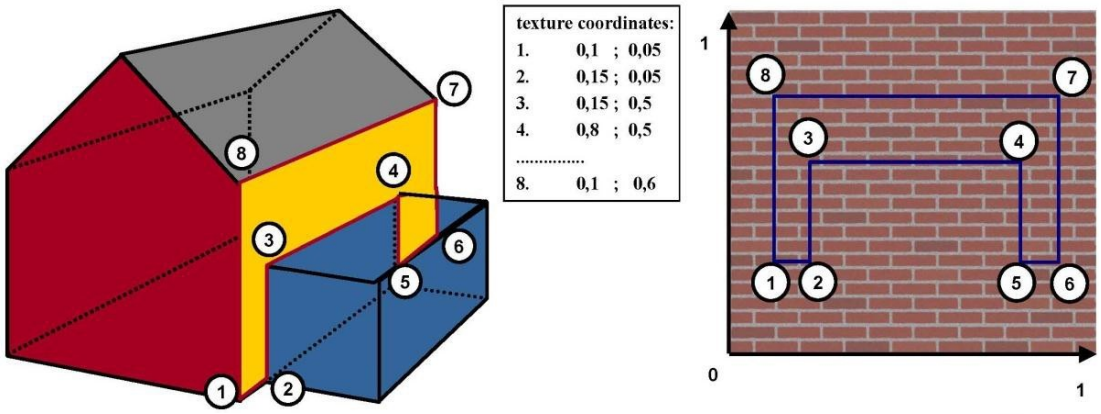

图 17. 使用纹理坐标对纹理进行定位（图片：IGG Uni Bonn）。

纹理坐标只适用于多边形表面，其边界由*gml: LinearRing*描述（例如，*gml:Triangle*，*gml:Polygon*，或由*gml:Polygons*组成的*gml:MultiSurface*）。它们定义了表面顶点与纹理空间中的点的明确映射，即每个顶点包括内环顶点都必须在纹理空间中得到一个相应的坐标对（关于坐标的概念，参考ISO 19111）。这些坐标不限于[0,1]区间。内部面点的纹理坐标是由顶点的纹理坐标平面插值而来。图16显示了一个例子。

目标表面几何对象的纹理坐标是使用*TexCoordList*作为*target*属性中的纹理参数化对象来指定的。每个外部和内部的*gml:LinearRing*构成了目标表面几何对象的边界（它也可能是一个*gml: CompositeSurface*，*gml:MultiSurface*，或*gml:TriangulatedSurface*）需要的纹理坐标集。使用*TexCoordList*的*textureCoordinates*元素来指定一组纹理坐标。因此，*TexCoordList*包含的*textureCoordinate*元素数量与目标面几何对象包含的*gml: LinearRings*一样多。*textureCoordinate*的强制性属性*ring*提供了各自环的*gml:id*。内容是一个有序的双值列表，每两个值定义一个纹理坐标对$(s,t)^T$，*s*表示水平纹理轴，*t*表示垂直纹理轴。列表中每个环点包含一个坐标对，坐标对的顺序与CityGML文件中环点的顺序相对应（不考虑可能翻转的面方向）。如果目标表面几何对象的存在环点没有被分配纹理坐标，映射就不完整，相应的表面就不能被纹理化。如果是聚合的目标几何对象，映射的完整性只决定于叶几何对象。

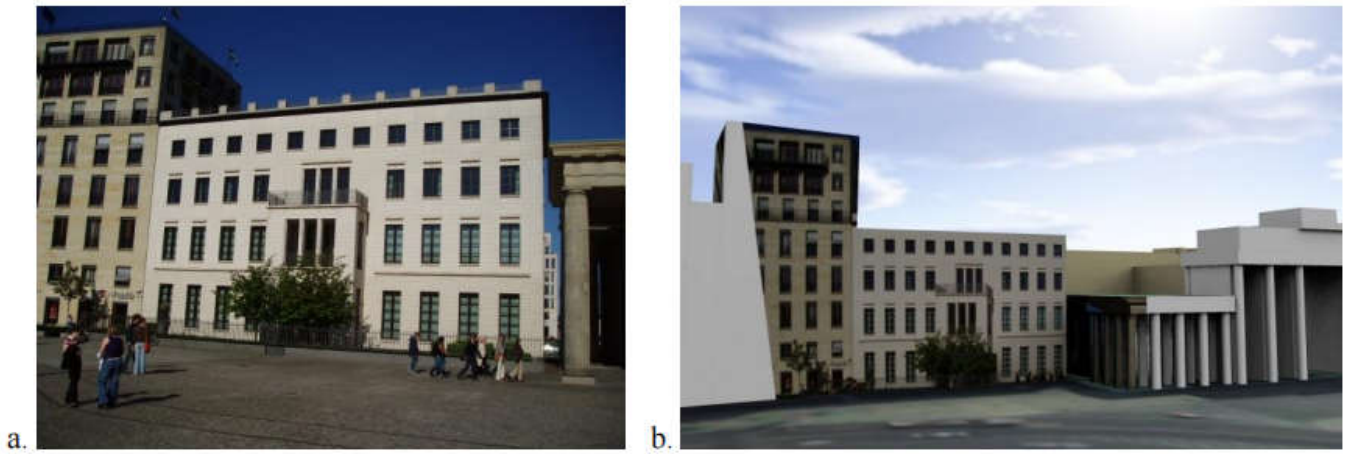

图 18. 使用worldToTexture变换将一张照片（a）投射到多个外墙（b）。照片并没有完全覆盖左边的外墙。因此，纹理似乎被剪切了。纹理包装被设置为 "none"（来源：柏林参议院，Hasso-PlattnerInstitute）。

另外，映射函数可以包括一个由*TexCoordGen*类指定的3x4转换矩阵。由*worldToTexture*元素指定的转换矩阵，定义了从齐次坐标的空间位置到纹理空间的线性转换。齐次坐标的使用有利于透视投影的转换，例如将一张照片投影到一个城市模型中（参见图18）。纹理坐标$(s,t)^T$从空间位置$(x,y,z)^T$计算为$(s,t)^T=(s'/q',t'/q')^T$与$(s',t',q')^T=M(x,y,z,1)^T$。*M*表示3x4变换矩阵。与一般的4x4变换相比，产生的*Z*分量被忽略了。因此，相应的矩阵行被省略了。此外，*worldToTexture*元素使用*gml:SRSReferenceGroup*属性来定义其CRS。世界空间中的

位置必须首先转换为这个CRS，然后才能应用变换矩阵。

下面的结构会产生一个*worldToTexture*变换，通过将世界空间（城市模型）中的一个位置投影到纹理空间中的一个位置，来模仿拍摄照片的过程。

$$M = \underbrace{\begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}}_{Adjustment\,to\,texture\,space} \cdot \underbrace{\begin{pmatrix} 2f/w & 0 & 0 & 0 \\ 0 & 2f/h & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{Perspective\,projection} \cdot \underbrace{\begin{pmatrix} r_x & r_y & r_z & 0 \\ u_x & u_y & u_z & 0 \\ d_x & d_y & d_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Camera\,orientation} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Camera\,location}$$

在这个公式中，*f*表示焦距；*w*和*h*表示图像传感器的物理尺寸；$\vec{r}$、$\vec{u}$和$\vec{d}$定义了相机的参照系，即用世界坐标表示的右、上和方向单位向量；*P*代表相机在世界空间的位置。图19描述了这种设置。

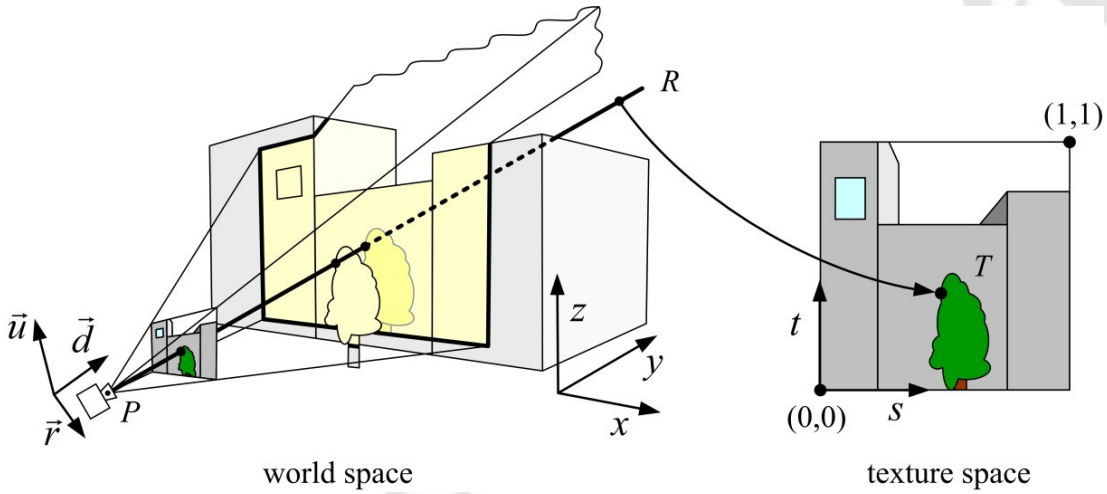

图 19. 投影式纹理映射。从投影中心P发出的射线R上的所有点都被映射到纹理空间中的同一个点T上（图片：Hasso-Plattner-Institute, IGG TU Berlin)。

另外，如果3x4相机矩阵M_P是已知的（例如通过校准和注册过程），它可以很容易地被用于*worldToTexture*。M_P是由内在和外在的相机参数定义的（内部和外部方向），并将世界空间中的一个位置转换为图像中的一个像素位置。假设图像左上角的像素坐标为（0,0），那么到纹理空间坐标的完整转换可以写成（$width_{image}$和$height_{image}$表示图像尺寸的像素）：

$$M = \begin{pmatrix} 1/width_{image} & 0 & 0 \\ 0 & -1/height_{image} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot M_P$$

请注意，*worldToTexture*不能补偿由真实相机镜头引入的径向或其他非线性失真。 *worldToTexture*的另一个用途是为一个具有复杂几何形状的外墙贴图，而不需要为每个*gml:LinearRing*指定贴图坐标。相反，只有使用*TexCoordGen*作为参数化，外墙的聚合面成为纹理目标。然后，*worldToTexture*有效地编码了世界空间到纹理空间的正射投影。对于垂直外墙的特殊情况，转换公式如下：

$$M = \underbrace{\begin{pmatrix} 1/width_f & 0 & 0 & 0 \\ 0 & 1/height_f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Scaling\,to\,texture\,space} \cdot \underbrace{\begin{pmatrix} -n_y & n_x & 0 & 0 \\ 0 & 0 & 1 & 0 \\ n_x & n_y & 0 & 0 \end{pmatrix}}_{Facade\,orientation} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & -F_x \\ 0 & 1 & 0 & -F_y \\ 0 & 0 & 1 & -F_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Facade\,location}$$

这个方程假设表示外墙$\vec{n}$的整体法向量（归一化，指向外侧，与地面平行），*F*表示外墙的左下角，$width_f$

和*height*$_f$指定外墙的世界单位尺寸。对于任意法向量，外墙方向矩阵的形式与摄像机方向矩阵相似。

$$M = \begin{pmatrix} 1/width_f & 0 & 0 & 0 \\ 0 & 1/height_f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_x & r_y & r_z & 0 \\ u_x & u_y & u_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -F_x \\ 0 & 1 & 0 & -F_y \\ 0 & 0 & 1 & -F_z \\ 0 & 0 & 0 & 1 \end{pmatrix} with \begin{cases} \vec{r} = \frac{(0,\ 0,\ 1)^T \times \vec{n}}{\| (0,\ 0,\ 1)^T \times \vec{n} \|} \\ \vec{u} = \vec{n} \times \vec{r} \end{cases}$$

ParameterizedTextureType, ParameterizedTexture, TextureAssociationType

```
<xs:complexType name="ParameterizedTextureType">
<xs:complexContent>
<xs:extension base="AbstractTextureType">
<xs:sequence>
<xs:element name="target" type="TextureAssociationType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfParameterizedTexture"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="ParameterizedTexture" type="ParameterizedTextureType"
substitutionGroup="_Texture"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfParameterizedTexture" type
="xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="TextureAssociationType">
<xs:sequence minOccurs="0">
<xs:element ref="_TextureParameterization"/>
</xs:sequence>
<xs:attribute name="uri" type="xs:anyURI" use="required"/>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

AbstractTextureParameterizationType, TexCoordListType, TexCoordGenType

```
<xs:complexType name="AbstractTextureParameterizationType" abstract="true
">
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType">
```

```
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTextureParameterization"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_TextureParameterization" type=
"AbstractTextureParameterizationType" abstract="true"
substitutionGroup="gml:_GML"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTextureParameterization"
type="xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="TexCoordListType">
<xs:complexContent>
<xs:extension base="AbstractTextureParameterizationType">
<xs:sequence>
<xs:element name="textureCoordinates" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="gml:doubleList">
<xs:attribute name="ring" type="xs:anyURI" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element ref="_GenericApplicationPropertyOfTexCoordList" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="TexCoordList" type="TexCoordListType" substitutionGroup
="_TextureParameterization"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTexCoordList" type=
```

```
 "xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="TexCoordGenType">
<xs:complexContent>
<xs:extension base="AbstractTextureParameterizationType">
<xs:sequence>
<xs:element name="worldToTexture">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="core:TransformationMatrix3x4Type">
<xs:attributeGroup ref="gml:SRSReferenceGroup"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element ref="_GenericApplicationPropertyOfTexCoordGen" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="TexCoordGen" type="TexCoordGenType" substitutionGroup=
"_TextureParameterization"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfTexCoordGen" type=
"xs:anyType" abstract="true"/>
```

9.5. 相关概念

外观的概念与通用覆盖方法有关（参见ISO 19123和OGC摘要规范，主题6）。表面数据可以被描述为具有特定的映射函数的二维域在面上的离散或连续覆盖。需要通过适当的映射函数和专业化的有效域和范围集扩展GML的覆盖范围（截至3.1版）。出于实施和使用上的简单性和易理解性的考虑，CityGML没有采用这种方法，而是基于纹理和材质描述计算机图形领域中的面属性（参见X3D、COLLADA规范、Foley等）。纹理和材质使用适当的映射将数据存储为颜色。如果这样的映射不切实际，可以用ADE来定制数据存储。在CityGML 2.0.0版本之后，将考虑对外观建模的覆盖范围进行审查。

外观也与描绘有关。描绘表达了数字模型图像的构成和符号化，即表现形式；而外观则编码了对真实物体表面的观测，即数据。即使是基于纹理和材质等图形术语，面数据也不只作为描绘的输入，同样也可以作为要素面分析的输入或输出。因此，CityGML并没有为描绘的目的定义主题的组合。描绘是留给查看器应用程序或样式规范语言，如OGC图层样式注记（SLD）或OGC符号学编码（SE）。

9.6. 代码列表

特征*_Texture*的*mimeType*属性被指定为*gml:CodeType*。这个属性的值可以在一个代码列表中列举出来。代码表的建议可以在附件C.6中找到。

9.7. 一致性要求

基本要求

1. 一个面几何对象在每个主题下最多可以成为两个纹理和两个材料（分别用于正面和背面）的目标。
2. *GeoreferencedTexture*元素的*referencePoint*属性（类型：*gml:PointPropertyType*）只能包含或引用一个具有2D坐标值的点几何对象。
3. *TexCoordList*元素的*textureCoordinates*属性定义了表面的边界点到纹理空间中的明确映射。表面的每个边界点都必须在纹理空间中有一个相应的坐标对。纹理空间中的坐标对应以每个边界点的两个双精度数字形式给出。坐标对的顺序必须遵循CityGML文件中边界点的顺序（不管表面方向是否可能翻转）。每个组成目标表面几何对象边界的*gml:LinearRing*需要它自己的纹理坐标集。
4. 一个*GeoreferencedTexture*元素必须提供内部或外部的地理参考，否则它是无效的。内部地理参考应通过*GeoreferencedTexture*元素的*referencePoint*属性（类型：*gml:PointPropertyType*）和orientation属性（类型：*core:TransformationMatrix2x2Type*）来声明。外部地理参考可以由纹理图像文件本身（例如GeoTIFF）或随附的世界文件提供。

参考完整性

1. *appearanceMember*元素（类型：*AppearancePropertyType*）可以包含一个内联的*Appearance*元素，或者使用GML 3.1.1的XLink概念引用一个远程*Appearance*元素。在后一种情况下，*appearanceMember*元素的*xlink:href*属性只能指向一个远程*Appearance*元素（其中远程*Appearance*元素位于另一个文档或同一文档的其他地方）。必须给出所包含的元素或引用，但既不能同时包含元素或引用，也不能完全不包含元素或引用。
2. *core:_CityObject*元素的*appearance*属性（类型：*AppearancePropertyType*）可以包含一个内联的*Appearance*元素，或者使用GML3.1.1的XLink概念引用一个远程*外观*元素。在后一种情况下，*外观*属性的*xlink:href*属性只能指向一个远程的*appearance*元素（其中远程的*Appearance*元素位于另一个文档中或同一文档的其他地方）。所包含的元素或引用必须给出，但既不能同时包含元素或引用，也不能完全不包含元素或引用。
3. *Appearance*元素的*surfaceDataMember*属性（类型：*SurfaceDataPropertyType*）可以包含一个内联的*_SurfaceData*元素，或者使用GML3.1.1的XLink概念引用一个远程的*_SurfaceData*元素。在后一种情况下，表面数据成员的*xlink:href*属性只能指向一个远程的*_SurfaceData*元素（其中远程的*_SurfaceData*元素位于另一个文档中或同一文档的其他地方）。无论是包含的元素还是引用都必须给出，但既不能同时包含元素或引用，也不能完全不包含元素或引用。
4. *ParameterizedTexture*元素的*target*属性（类型：*TextureAssociationType*）可以包含一个内联的*_TextureParameterization* 元素，或者使用GML 3.1.1的XLink概念引用一个对远程*_TextureParameterization*元素。在后一种情况下，*target*属性的*xlink:href*属性只能指向一个远程*_TextureParameterization*元素（其中远程*_TextureParameterization*元素位于另一个文档中或同一文档的其他地方）。必须给出所包含的元素或引用，但既不能同时包含元素或引用，也不能完全不包含元素或引用。

5. *GeoreferencedTexture*元素的*target*属性（类型为*xs:anyURI*）应指定目标面几何对象的*gml:id*，它只能是*gml:AbstractSurfaceType*或*gml:MultiSurface*类型。
6. 复合类型*TextureAssociationType*的*uri*属性应指定目标表面几何对象的*gml:id*，它只能是*gml:AbstractSurfaceType*或*gml:MultiSurface*类型。
7. *TexCoordList*元素的*textureCoordinates*属性的*ring*属性应指定目标表面几何对象的*gml:id*，它只能是*gml:LinearRing*类型。
8. *X3DMaterial*元素的*target*属性（类型为*xs:anyURI*）应指定目标表面几何对象的*gml:id*，它只能是*gml:AbstractSurfaceType*或*gml:MultiSurface*类型。

9.8. CityGML先前版本的材质模型[已弃用]

由于GML3没有内置的表面材质表示概念，CityGML以前的版本通过*TexturedSurface*类来扩展GML3的几何模型，它允许为三维面分配外观属性（颜色、亮度、透明度）和纹理。外观属性的定义是从X3D规范中采用的。由于固有的局限性，这种外观建模的方法已经被弃用了。然而，为了给现有的CityGML实现一定的向后兼容性，该方法已被纳入CityGML 1.0版本和2.0版本，成为一个单独的扩展模块*TexturedSurface*。通过这种方式，可以采用旧的材质模型支持这个模块。请注意，根据*TexturedSurface*模块建模的外观信息可以在没有信息损失的情况下转换为CityGML的*Appearance*模块概念，本章前面的条款已经介绍过。因此，**强烈不建议**使用*TexturedSurface*模块，应该坚持使用*Appearance*模块。此外，*TexturedSurface*模块预计将在CityGML的未来版本中被移除。

对于*TexturedSurface*模块，每个表面或复合面都可以被专门化为一个*TexturedSurface*，它可以被赋予*Materials*（*颜色*、*亮度*、*透明度*）或*SimpleTextures*。图20描述了UML图，关于XML模式的定义见附件A.14。

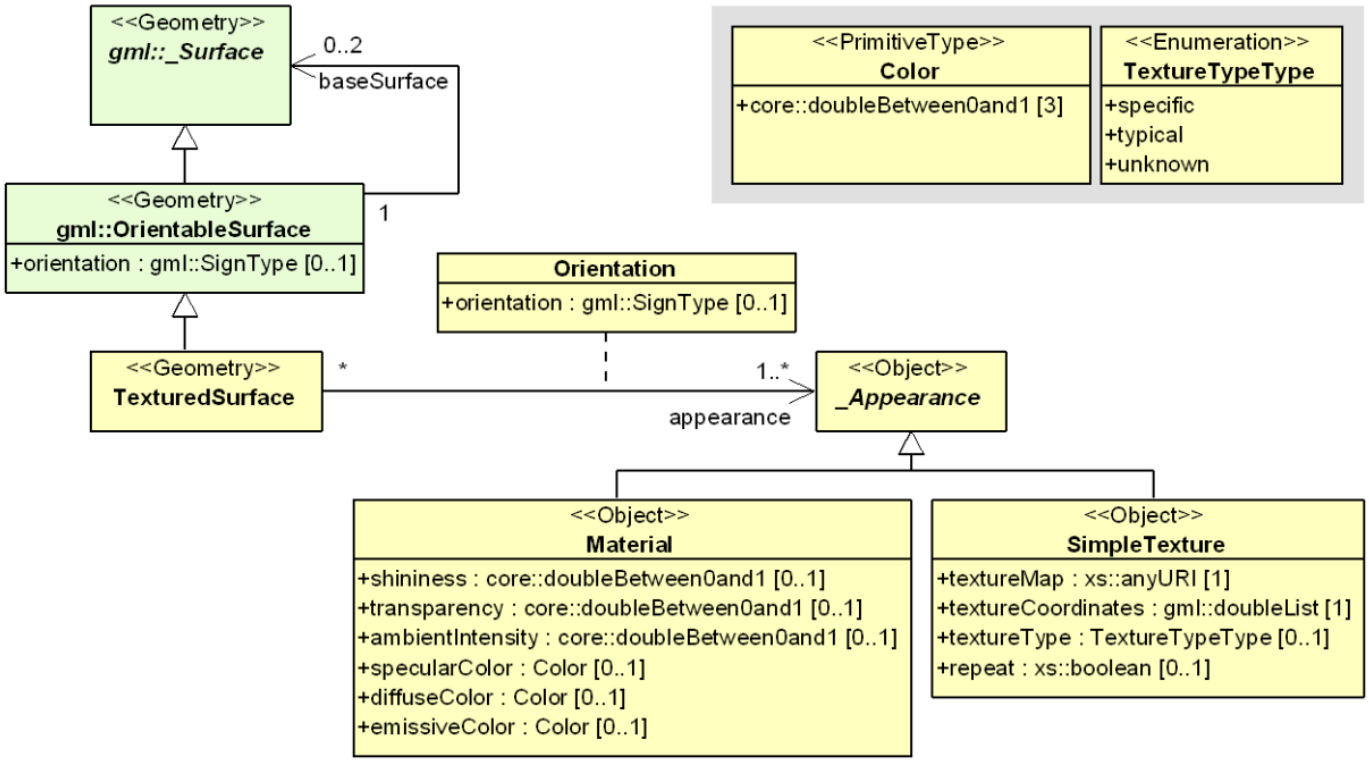

图 20. CityGML的材质模型的UML图。请注意，这种外观建模的方法已经被弃用，预计将在未来的CityGML版本中被删除。前缀用于指示与模型元素相关的 XML 命名空间。没有前缀的元素名称是在CityGML的TexturedSurface模块中定义的。

在面上定位纹理的概念符合三维计算机图形标准X3D（web 3D 2004），是VRML97的后继者。CityGML在GML3的几何模型中加入了*TexturedSurface*类，因为在ISO 19107和GML3中没有合适的纹理概念。

纹理被指定为由*URI*（统一资源标识符）引用的栅格图像，可以是一个任意的资源，包括在互联网上。纹理是通过采用*texture coordinates*的概念来定位的，即每个纹理坐标与*TexturedSurface*的一个三维坐标完全匹配（图17）。纹理坐标的使用允许在面几何形状上准确定位和调整纹理。

一个表面的颜色是由RGB值定义的。他们必须在0到1之间取值。*frontOpacity*和*backOpacity*分别定义每个面的*transparency*程度。它们的值也必须在0到1的范围内，其中1表示完全不透明，0表示完全透明。颜色可以区分为*diffuseColor*漫射色（被光源照射时的颜色）、*emissiveColor*自发光色（材质自身的颜色）和*specularColor/shininess*镜面色/亮度（表面的亮度）。

纹理可以由*textureType*属性来限定。*textureType*区分了特定对象的纹理（*specific*）和典型对象的纹理（*typical*）。纹理也可以被分类为*unknown*。

*_Appearance*是从*gml:AbstractGMLType*派生出来的，可以在一个*appearance*属性中被引用。属性*gml:id*是继承的，它的值可以被一个XLink引用。*_Appearance* 是材质*Material*和*SimpleTexture*的父类。

Xml命名空间

CityGML *TexturedSurface*模块的XML命名空间是由统一资源标识符（URI） http://www.opengis.net/citygml/texturedsurface/2.0 定义。在*TexturedSurface*模块的XML 模式定义中，这个URI也被用来标识默认命名空间。

9.8.1. 纹理表面

TexturedSurfaceType, TexturedSurface, AppearancePropertyType

```
<xs:complexType name="TexturedSurfaceType">
<xs:complexContent>
<xs:extension base="gml:OrientableSurfaceType">
<xs:sequence>
<xs:element ref="appearance" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="TexturedSurface" type="TexturedSurfaceType"
substitutionGroup="gml:OrientableSurface"/>
<!--
================================================================================
============== -->
<xs:element name="appearance" type="AppearancePropertyType"/>
<!--
================================================================================
============== -->
<xs:complexType name="AppearancePropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="_Appearance"/>
</xs:sequence>
<xs:attribute name="orientation" type="gml:SignType" default="+"/>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

*TexturedSurface*可以有一个或多个外观属性，它可以是一个材质*Material*（颜色，…）或一个*SimpleTexture*。*_Appearance*元素既可以作为这种类型的元素内联表示，也可以通过XLink引用一个远程的*_Appearance*元素。必须给出引用或包含的元素，但既不能同时包含元素或引用，也不能完全不包含元素或引用。*_Appearance*所指的面是由外观属性元素的*orientation*属性（类型*gml:SignType*）给出的，它对应*orientation*属性。+表示正方向的一面，-表示负方向的一面。

AbstractAppearanceType,__Appearance

```
<xs:complexType name="AbstractAppearanceType" abstract="true">
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType"/>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_Appearance" type="AbstractAppearanceType" abstract=
"true" substitutionGroup="gml:_GML"/>
```

MaterialType, Material

```
<xs:complexType name="MaterialType">
<xs:complexContent>
<xs:extension base="AbstractAppearanceType">
<xs:sequence>
<xs:element name="shininess" type="core:doubleBetween0and1" minOccurs="0
"/>
<xs:element name="transparency" type="core:doubleBetween0and1" minOccurs=
"0"/>
<xs:element name="ambientIntensity" type="core:doubleBetween0and1"
minOccurs="0"/>
<xs:element name="specularColor" type="Color" minOccurs="0"/>
<xs:element name="diffuseColor" type="Color" minOccurs="0"/>
<xs:element name="emissiveColor" type="Color" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Material" type="MaterialType" substitutionGroup=
"_Appearance"/>
```

SimpleTextureType, SimpleTexture, TextureTypeType

```
<xs:complexType name="SimpleTextureType">
<xs:complexContent>
<xs:extension base="AbstractAppearanceType">
<xs:sequence>
<xs:element name="textureMap" type="xs:anyURI"/>
<xs:element name="textureCoordinates" type="gml:doubleList"/>
<xs:element name="textureType" type="TextureTypeType" minOccurs="0"/>
<xs:element name="repeat" type="xs:boolean" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="SimpleTexture" type="SimpleTextureType"
substitutionGroup="_Appearance"/>
<!--
================================================================================
============== -->
<xs:simpleType name="TextureTypeType">
<xs:restriction base="xs:string">
<xs:enumeration value="specific"/>
<xs:enumeration value="typical"/>
<xs:enumeration value="unknown"/>
</xs:restriction>
</xs:simpleType>
```

9.8.2. 一致性要求

参照完整性

- *TexturedSurface*元素的*appearance*属性（类型:*AppearancePropertyType*）可能包含一个内联的*_Appearance*元素，或者一个使用GML 3.1.1的XLink概念的远程*_Appearance*元素。在后一种情况下，*外观*属性的*xlink:href*属性只能指向一个远程的*_Appearance*元素（其中远程的*_Appearance*元素位于另一个文档中或同一文档的其他地方）。必须给出所包含的元素或引用，但既不能同时包含元素或引用，也不能完全不包含元素或引用。

10. 主题模型

CityGML的主题模型，是由虚拟3D城市模型中最重要的对象类型定义而成。在许多不同的应用领域中，这些对象类型十分重要。大多数主题类（可传递地）都是由基本类 *Feature* 和 *FeatureCollection* 派生而来，这些基本概念在ISO 19109和GML3中定义，可用于表示单一空间对象和它们之间的聚合。特征包含空间属性和非空间属性，这些属性被映射到GML3的特征属性，并具有相应的数据类型。几何特性与第8章中描述的几何类互相关联。主题模型还包括不同类型的特征类之间的相互关系，如聚合，概括和关联。

显式建模的目的是在不同的应用程序之间实现高度的语义互操作性。通过指定主题概念及其语义以及它们到UML和GML3的映射，不同的应用程序可以依赖于一组定义良好的 *feature types* ，属性和数据类型。这些特征类型，属性和数据类型具有标准化的含义或解释。为了允许在CityGML中没有明确建模的对象和/或属性的交换，引入了 *generic city objects* 和 *attributes* 的概念以及CityGML的 *ADE* 机制（参见第10.12章和第10.13章）。

CityGML主题模型的每个字段都由一单独的个CityGML扩展模块进行解释。因此，扩展模块是通过分割整个CityGML主题数据模型而得到的。所有扩展模块都基于CityGML核心模块，并且依赖于CityGML核心模块。核心部分包括CityGML数据模型的基本概念和组件。实施者可以根据特定的信息需求或应用领域选择将CityGML扩展模块与核心部分结合起来。对于CityGML的2.0版本，定义了以下13个主题扩展模块：*外观，桥梁，建筑，城市家具，城市对象组，泛型，土地利用，地形，交通，隧道，植被，水体和纹理表面 [已弃用]* 。CityGML模块之间的有效组合被称为CityGML专用文件。通过这种方式，CityGML专用文件可以允许对整个CityGML数据模型进行部分实现（参见第7章）。

CityGML数据模型所涵盖的主题领域将在本章的子章节中介绍。每个子章节都与特定的CityGML模块相关。

10.1. CityGML核心模块

CityGML Core 模块定义了整个CityGML数据模型的基本概念和组件。它形成了CityGML数据模型的基础数据结构，也包括了所有扩展模块的依赖关系。因此，核心模块必须由任何符合要求的系统来实现。首先，核心模块提供了抽象的基类，扩展模块中的主题类是从这些抽象基类（可传递地）派生的。除抽象类外，核心还包含非抽象的内容，例如可由多个扩展模块使用的基本数据类型和主题类。图21中的UML图说明了CityGML的核心模块，XML模式定义见下文和附录A.1。

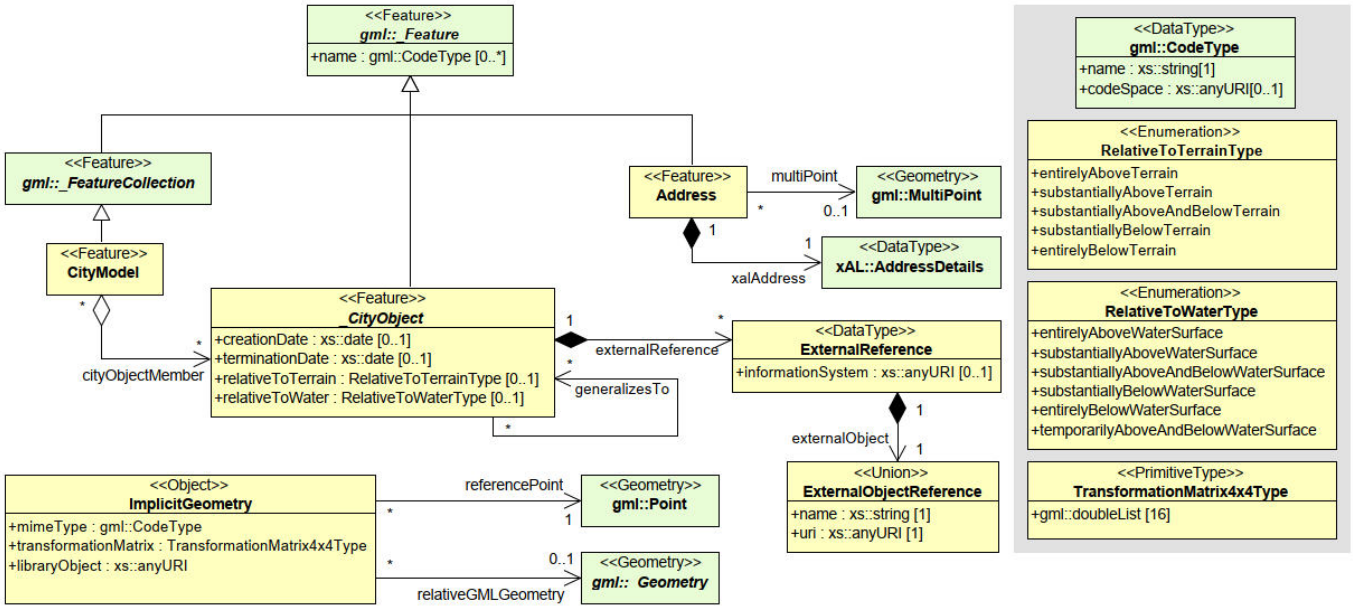

图 21. CityGML核心模块的UML图。属性名称括号中的数字，表示该属性的多重性：每个对象的属性其最小和最大出现次数。例如，名称在类功能中是可选的（0），或者可以出现多次（星号），城市对象可以出现最多一个 creationDate 。前缀用于指示与模型元素关联的XML名称空间。没有前缀的元素名在CityGML核心模块中定义）

CityGML数据模型中所有主题类的基类都是抽象类 *_CityObject*。 *_CityObject*提供了创建日期与终止日期，用于管理特征的历史记录，以及在其他数据集中对同一对象的外部引用。此外，还提供了两个定性属性 *relativeToTerrain* 和 *relativeToWater* ，用于指定特征相对于地形和水面的位置。其中的拓扑关系如图22所示。这两个属性都有助于简单而高效的查询，比如地下建筑物的数量（*entirelyBelowTerrain*），而不需要额外的数字地形模型或水体模型。

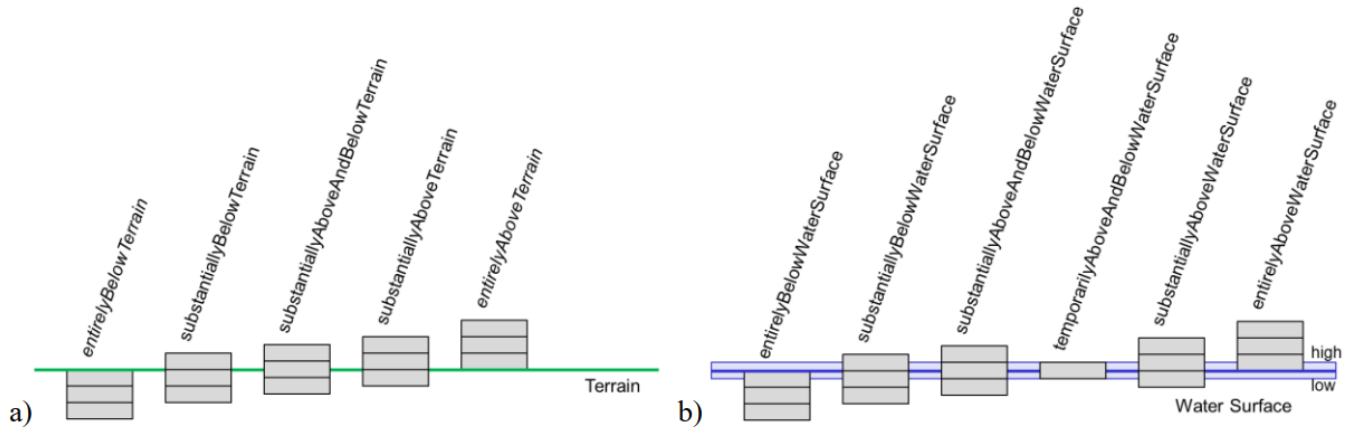

图 22. CityGML对象相对于 a) 地形和 b) 水面的拓扑关系。

_CityObject 是GML类 *_Feature* 的一个子类，因此它从 *_GML*继承元数据属性（例如，关于继承关系，质量，准确性，本地CRS的信息）和名称属性。一个 *_CityObject*可以有多个名称，这些名称可以由一个 *codeSpace* 进行限定。该功能可以用于如下场景，区分正式名称和流行名称或不同语言的名称（参见GML objects的name属性，Cox等人，2004）。 *_CityObject* 的泛化属性 *generalizesTo* 可用于关联特征，这些特征以不同的细节层次表示同一真实世界对象，即特征及其泛化对应物。这种关系的方向是从特征到相应的广义特征。

主题类可能有基于关系，属性和几何的继承子类。 *_CityObject* 专用子类的特征可以聚合到单个 *CityModel*，这是一个具有可选元数据的特征集合。除非另有说明，通常，每个特性都有属性 *class, function, 和usage* 。*class* 属性只能出现一次，而 *usage* 和 *function* 属性可以使用多次。*class* 属性允许对 *_CityObject* 主题类层次结构之外的要素进行分类。例如，建筑特征由主题子类表示建筑：建筑_首先是 *_CityObject*（该子类在CityGML的构建模块中定义，参见第10.3章）。进一步的分类，例如住宅或行政大楼，可以使用类的*class*属性进行建模建筑：建筑。属性函数通常表示对象的预期用途或用途，例如建筑物的酒店或购物中心，而属性用途通常定义其实际用途。attrib utes*class*, *function* 和 *usage* 的可能值可以在代码列表中指定，建议按照GML 3.1.1的简单字典专用文件（参见第6.6章和第10.14章）将代码列表实现为简单字典。附录C提供了由SIG 3D提出和维护的代码列表，其中包含可行的属性值，用户可以扩展或重新定义这些属性值。

除了主题内容外，核心模块还提供了隐式几何的概念，作为GML3几何模型的增强。由于这一概念与CityGML的空间模型密切相关，因此已在第8.2章中介绍。

CityGML中主题模型的顶级类层次结构如图23所示。 *_CityObject* 的子类包括城市模型的不同主题字段，这些字段由单独的CityGML扩展模块覆盖：地形，建筑物，桥梁，隧道，土地使用对象，水体，植被，一般城市对象，城市家具对象，城市对象组和交通。为了指示定义 *_CityObject* 的相应子类的扩展模块，图23中的类名前面有前缀。每个前缀都与一个CityGML扩展模块相关联（参见第4.3章和第7章了解CityGML扩展模块和相应前缀的列表）。

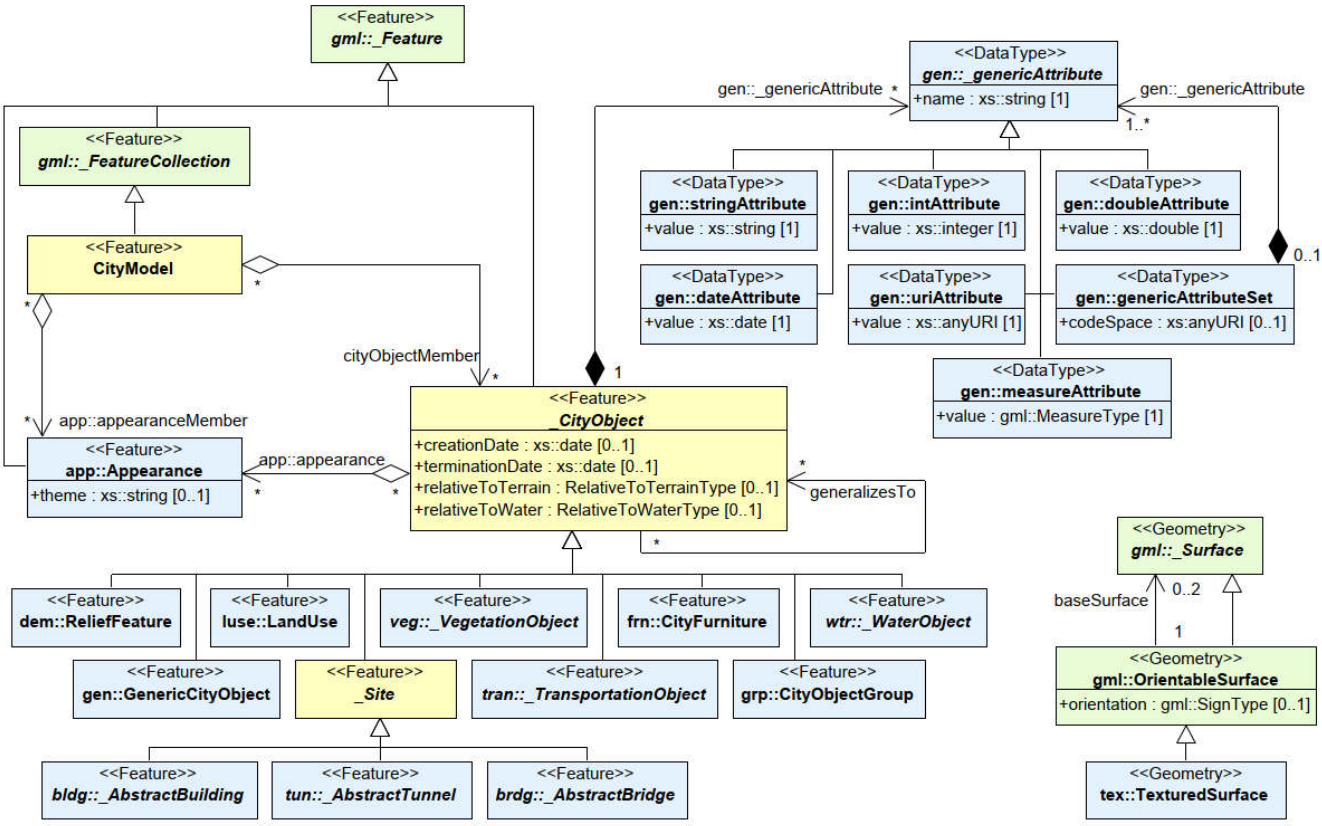

图 23. CityGML的顶级类层次结构。前缀用于指示与模型元素关联的XML名称空间。没有前缀的元素名称在CityGML核心模块中定义。

CityGML的泛型模块（参见第6.11和10.12章）中定义的类*GenericCityObject* 和 *_genericAttribute*，允许对任何其他主题类未涵盖或在CityGML中未表示属性的3D对象进行建模和交换。例如，在将来，从核心模块的抽象类 *_Site*中派生的地块可能会由其他子类加以补充，如发掘，城墙或路堤。目前，为了表示和交换这些特性，应该使用类 *GenericCityObject*。但是，只有在任何其他CityGML模块未提供适当的主题类或属性的情况下，才能使用通用城市对象和属性的概念。

如果使用*Generics* 模块，则可以为每个 *CityObject* 分配任意数量的泛型属性，用来表示特征的附加属性。为此，*Generics*模块通过属性元素 *_genericAttribute*扩充抽象基类 *_CityObject*。使用CityGML的应用领域扩展机制将附加属性 *_genericAttribute*注入到 *_CityObject*中（参见第10.13章）。通过这种方式， *_CityObject* 的每个主题子类都继承了这个属性，因此也就都可以包含泛型属性。因此，泛型模块对定义 *_CityObject*的主题子类中，所有CityGML扩展模块都有一定的影响。

关于特征曲面的外观信息可以由CityGML的外观模块提供的类 *Appearance* 来表示（参见第9章）。与核心的其他主题扩展不同的是，它不是从 *CityObject 派生出来的，而是从GML类 _Feature* 派生出来的。— *_CityObject* 要素和*Apperance* 要素可以使用 *CityObjectMember* 和*appearanceMember* 以任意顺序包含在*CityModel* 要素集合中 （参见第9章和第10.1.1章）。此外，外观可以存储在 *_CityObject* 本身的内联中。为了使城市对象能够存储外观信息，外观模块使用CityGML的应用领域扩展机制，通过属性元素外观来扩充抽象基类 *_CityObject*（参见第10.13章）。因此，只有在支持*Apperance* 模块的情况下，*Apperance* 属性才能对_CityObject及其主题子类可用。所以，像通用模块一样，*Apperance* 模块对任何其他扩展模块都有直接影响。

为了完整性，图23中还示出了 *TexturedSurface*。该外观建模方法已经被弃用，并会在将来的CityGML版本中被删除。由于 *TexturedSurface* 所包含的信息可以无损地转换到*Appearance* 模块，因此强烈建议不要使用*TexturedSurface*。

XML 命名空间

CityGML *CityGML Core*模块的XML命名空间由统一资源标识符（URI）标识http://www.opengis.net/citygml/relief/2.0. 在*Relief*模块的xmlschema定义中，这个URI还用于标识默认名称空间。

10.1.1. 基本元素

Abstract *CityObjectType，_CityObject*

```
<xs:complexType name="AbstractCityObjectType" abstract="true">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="creationDate" type="xs:date" minOccurs=
"0"/>
                <xs:element name="terminationDate" type="xs:date"
minOccurs="0"/>
                <xs:element name="externalReference" type=
"ExternalReferenceType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="generalizesTo" type=
"GeneralizationRelationType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="relativeToTerrain" type=
"RelativeToTerrainType" minOccurs="0"/>
                <xs:element name="relativeToWater" type=
"RelativeToWaterType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfCityObject"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_CityObject" type="AbstractCityObjectType" abstract=
"true" substitutionGroup="gml:_Feature"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityObject" type=
"xs:anyType" abstract="true"/>
```

CityModelType, CityModel

```
<xs:complexType name="CityModelType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureCollectionType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfCityModel"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="CityModel" type="CityModelType" substitutionGroup=
"gml:_FeatureCollection"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityModel" type="
xs:anyType" abstract="true"/>
```

_*CityObject*Member

```
<xs:element name="cityObjectMember" type="gml:FeaturePropertyType"
substitutionGroup="gml:featureMember"/>
```

AbstractSiteType, _Site

```
<xs:complexType name="AbstractSiteType" abstract="true">
    <xs:complexContent>
        <xs:extension base="AbstractCityObjectType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfSite"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
================
-->
<xs:element name="_Site" type="AbstractSiteType" abstract="true"
substitutionGroup="_CityObject"/>
<!--
================================================================================
================
-->
<xs:element name="_GenericApplicationPropertyOfSite" type="xs:anyType"
abstract="true"/>
```

抽象类 *_site*是建筑物，桥梁，隧道，设施等的父类。CityGML未来的扩展（如发掘，城墙或路堤）将被建模为*_site*的子类。作为*CityObject*的子类， *_Site* 继承所有属性和关系，特别是id，名称，外部引用和泛化关系。

10.1.2. Generalisation relation, RelativeToTerrainType 和 RelativeToWaterType

GeneralizationRelationType

```
<xs:complexType name="GeneralizationRelationType">
    <xs:sequence minOccurs="0">
        <xs:element ref="_CityObject"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

RelativeToTerrainType, RelativeToWaterType

```
<xs:simpleType name="RelativeToTerrainType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="entirelyAboveTerrain"/>
        <xs:enumeration value="substantiallyAboveTerrain"/>
        <xs:enumeration value="substantiallyAboveAndBelowTerrain"/>
        <xs:enumeration value="substantiallyBelowTerrain"/>
        <xs:enumeration value="entirelyBelowTerrain"/>
    </xs:restriction>
</xs:simpleType><!--
================================================================================
============== -->
<xs:simpleType name="RelativeToWaterType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="entirelyAboveWaterSurface"/>
        <xs:enumeration value="substantiallyAboveWaterSurface"/>
        <xs:enumeration value="substantiallyAboveAndBelowWaterSurface"/>
        <xs:enumeration value="substantiallyBelowWaterSurface"/>
        <xs:enumeration value="entirelyBelowWaterSurface"/>
        <xs:enumeration value="temporarilyAboveAndBelowWaterSurface"/>
    </xs:restriction>
</xs:simpleType>
```

10.1.3. 外部引用

ExternalReference 定义了从 *_CityObject* 到另一个信息系统中相应对象的超链接，例如在ALKIS，ATKIS或OS MasterMap中。 引用由外部信息系统的名称（用URI表示）和外部对象的引用（用字符串或URI表示）组成。如果*ExternalReference* 中缺少*informationSystem*元素，则*ExternalObjectReference* 必须是URI。

ExternalReferenceType, ExternalObjectReferenceType

```
<xs:complexType name="ExternalReferenceType">
    <xs:sequence>
        <xs:element name="informationSystem" type="xs:anyURI" minOccurs="
0"/>
        <xs:element name="externalObject" type=
"ExternalObjectReferenceType"/>
    </xs:sequence>
</xs:complexType><!--
================================================================================
================ -->
<xs:complexType name="ExternalObjectReferenceType">
    <xs:choice>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="uri" type="xs:anyURI"/>
    </xs:choice>
</xs:complexType>
```

10.1.4. 地址信息

CityGML核心模块提供了在虚拟城市模型中表示真实世界要素地址信息的方法。由于并非每个真实世界的功能都被分配了地址，因此没有为基类 *_CityObject*定义相应的*address*属性，但如果需要，则可以在其子类中对地址进行声明。例如，建筑模型为其子类 *_AbstractBuilding*和*Door*声明*address*属性。两个类都引用核心模块的相应数据类型来表示地址信息（参见第10.3章）。

地址信息具有xalAddress属性和multiPoint(可选择)属性。例如，对于建筑要素，*multiPoint*特性允许指定与相应地址关联的建筑入口位置。点坐标可以是2D或3D。将地址建模为特征有一个优点，即GML3可以使用XLinks对特征进行引用。这意味着，地址可能被存储在外部文件中，或可被外部Web进行调用。CityGML文件中的地址属性元素不包含内联地址信息，只包含对相应外部特征的引用。

地址信息是使用oasis2003发布的*xALaddress*标准指定，该标准为各种国际地址提供了通用模式。因此，*Address*的*xalAddress*属性的子元素必须根据OASIS xAL模式进行构造。

AddressPropertyType, AddressType, Address

```
<xs:complexType name="AddressPropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="Address"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType><!--
================================================================================
================ -->
<xs:complexType name="AddressType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="xalAddress" type="
xalAddressPropertyType"/>
                <xs:element name="multiPoint" type=
"gml:MultiPointPropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfAddress"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="Address" type="AddressType" substitutionGroup=
"gml:_Feature"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfAddress" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="xalAddressPropertyType">
    <xs:sequence>
        <xs:element ref="xAL:AddressDetails"/>
    </xs:sequence>
</xs:complexType>
```

以下两个CityGML数据集的摘录包含用xAL表示德语和英语地址的示例。根据CityGML *Buildings*模块（参见第10.3章），地址信息附加到建筑对象(*bldg:Building*)。通常，如果CityGML实例文档包含地址信息，则命名空间前缀 “xAL” 应在根元素中声明，并且必须引用urn:oasis:names:tc:ciq:xsdschema：xAL:2.0。附录G.1中提供了一个完整的CityGML数据集示例，其中包含带有地址元素的建筑物。

```
<bldg:Building> …    <bldg:address>
        <core:Address>
            <core:xalAddress>                <!-- Bussardweg 7, 76356
Weingarten, Germany -->
                <xAL:AddressDetails>
                    <xAL:Country>
                        <xAL:CountryName>Germany</xAL:CountryName>
                        <xAL:Locality Type="City">
                            <xAL:LocalityName>
Weingarten</xAL:LocalityName>
                            <xAL:Thoroughfare Type="Street">
                                <xAL:ThoroughfareNumber>
7</xAL:ThoroughfareNumber>
                                <xAL:ThoroughfareName>
Bussardweg</xAL:ThoroughfareName>
                            </xAL:Thoroughfare>
                            <xAL:PostalCode>
                                <xAL:PostalCodeNumber>
76356</xAL:PostalCodeNumber>
                            </xAL:PostalCode>
                        </xAL:Locality>
                    </xAL:Country>
                </xAL:AddressDetails>
            </core:xalAddress>
        </core:Address>
    </bldg:address>
</bldg:Building>
```

```
<bldg:Building> …    <bldg:address>
        <core:Address>
            <core:xalAddress>                <!-- 46 Brynmaer Road
Battersea LONDON, SW11 4EW United Kingdom -->                <!-- source:
http://xml.coverpages.org/xnal.html -->
                <xAL:AddressDetails>
                    <xAL:Country>
                        <xAL:CountryName>United Kingdom</xAL:CountryName>
                        <xAL:Locality Type="City">
                            <xAL:LocalityName>LONDON</xAL:LocalityName>
                            <xAL:DependentLocality Type="District">
                                <xAL:DependentLocalityName>
Battersea</xAL:DependentLocalityName>
                                <xAL:Thoroughfare>
                                    <xAL:ThoroughfareNumber>
46</xAL:ThoroughfareNumber>
                                    <xAL:ThoroughfareName>Brynmaer
Road</xAL:ThoroughfareName>
                                </xAL:Thoroughfare>
                            </xAL:DependentLocality>
                            <xAL:PostalCode>
                                <xAL:PostalCodeNumber>SW11
4EW</xAL:PostalCodeNumber>
                            </xAL:PostalCode>
                        </xAL:Locality>
                    </xAL:Country>
                </xAL:AddressDetails>
            </core:xalAddress>
        </core:Address>
    </bldg:address>
</bldg:Building>
```

10.1.5. 代码列表

*ImplicitGeometry*的*mimeType*属性指定为*gml:CodeType*. 此属性的值可以在代码列表中被枚举。这个代码列表的建议规范可以在附件C.6中找到。

10.1.6. 一致性要求

基础要求

1. *CityModel*元素（类型：*CityModelType*，替换组：*gml : FeatureCollection*）只能包含*cityObjectMember*元素（类型：*gml:FeaturePropertyType*), *app:appearanceMember*元素 （类型：app:*AppearancePropertyType*)，和*gml:featureMember*元素（类型：*gml:FeaturePropertyType*)作为功能成员。
2. *ExternalObjectReference*引入了两个元素*name*（类型：*xs:string*）和*uri*（类型：*xs:anyURI*）。外部引用可以由它们中的任何一个指定。但是，如果没有提供*ExternalReferenceType*类型的*informationSystem*

属性元素（类型：*xs:anyURI*），则必须提供*ExternalObjectReference*的*uri*元素。

3. 为了表示某个特征的地址信息，该特征的相应主题类应定义*AddressPropertyType*类型的属性。因此，对于所有CityGML扩展模块，只有*AddressPropertyType*类型应用于提供地址信息的元素。

4. 由于隐式几何的概念（参见第8.2章）是*CityGML Core*模块的一部分，因此为隐式几何引入的一致性要求（参见第8.3.3章）是核心一致性要求的一部分。

引用一致性

5. *_CityObjectMember*元素（类型：*gml:FeaturePropertyType*)可能包含一个 *_CityObject*元素，该元素通常是派生子类的对象，如*bldg:Building*，使用GML 3.1.1中的XLink概念，对远程 *_CityObject* 的内联或XLink引用。在后一种情况下 *_CityObjectMember*的*XLink:href* 属性只能指向远程 *_CityObject*元素（其中远程 *_CityObject*元素位于另一个文档或同一文档中的其他位置）。必须给出包含的元素或引用，但不能同时给出或不给出。

6. *AddressPropertyType*类型可以包含*Address*元素内联或使用gml3.1.1的远程*Address*元素XLink引用。在后一种情况下，*AddressPropertyType*的*XLink:href* 属性，其对应元素只能指向远程地址元素（其中远程地址元素位于另一个文档或同一文档的其他位置）。必须给出包含的元素或引用，但不能同时给出或不给出。

10.2. 数字地形模型（DTM）

城市模型的一个重要部分是地形。CityGML的数字地形模型（DTM）由专题扩展模块*Relief*提供（参见第7章）。在CityGML中，地形由lod0-4中的类*relief-feature*表示（图24描绘了UML图，XML模式定义见附录A.9）。*ReliefFeature*由*ReliefComponent*的一个或多个实体组成。其有效性可以限制在由可选有效范围多边形定义的特定区域。由于*ReliefFeature*和*ReliefComponent*继承自 *_CityObejct*，因此相应的属性和关系也被继承下来。*ReliefFeature*可以与不同概念的地形表示进行关联。地形可以指定为规则光栅或栅格（*RasterRelief*），三角网（*TINReflief*），打断线（*BreaklineRelief*）或质点（*MasspointRelief*）。这四种类型由相应的GML3类实现：*gml:RectifiedGridCoverage*实现栅格图， *gml:MultiCurve*可实现打断线， *gml:MultiPoint* 实现质量点类型，*gml:TriangulatedSurface / gml:Tin*都可构成三角网数据。如果在*gml:TriangulatedSurfaces*中，显示表达了三角面，同时*gml:Tin*中 只表示三维点，这时其中三角剖分可通过标准方法重建（Delaunay三角剖分，参见Okabe et al.1992）。打断线由三维曲线表示。质点只是一组三维点。

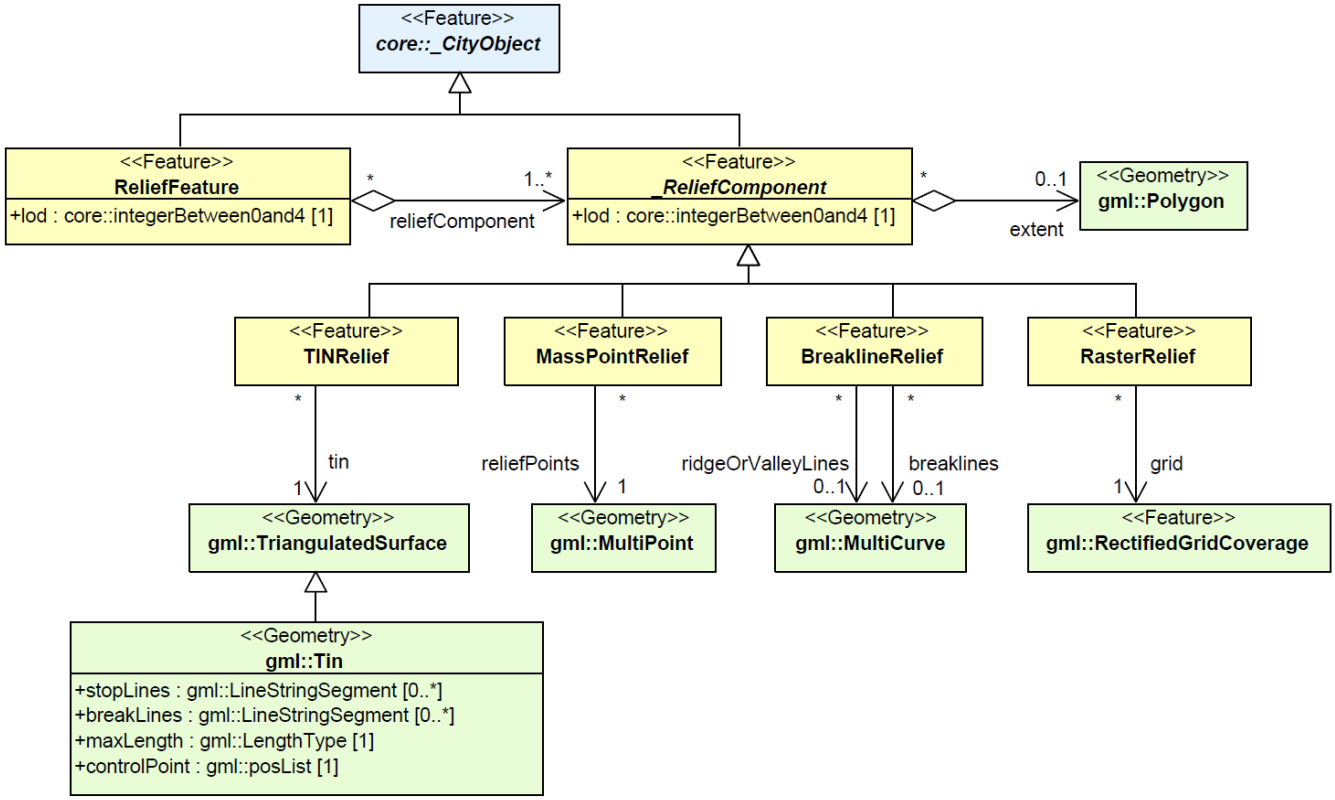

图 24. CityGML中数字地形模型的UML图。前缀用于指示与模型元素关联的XML名称空间。没有前缀的元素名称在CityGML Relief模块中定义。

在CityGML数据集中，四种地形类型可以以不同的方式组合在一起，产生了很高的灵活性。首先，每种类型可以用不同的细节级别来表示，反映出不同的精确度或分辨率。第二，可以通过多种类型的组合来描述地形的一部分，例如通过光栅和打断线，或者通过三角网和打断线。在这种情况下，打断线必须与三角形共享几何图形。第三，相邻区域可以用不同类型的地形模型来表示。为了便于这种组合，为每个地形对象提供了表示其有效性范围的*空间属性*（图25）。在大多数情况下，常规光栅数据集的有效范围与其边界框相对应。此有效范围由二维示意图多边形表示，该多边形可能有孔。例如，这一概念可以通过粗网格对地形进行建模，其中一些显著区域由详细的高精度TIN表示。两种类型之间的边界由相应地形对象的范围属性给出。

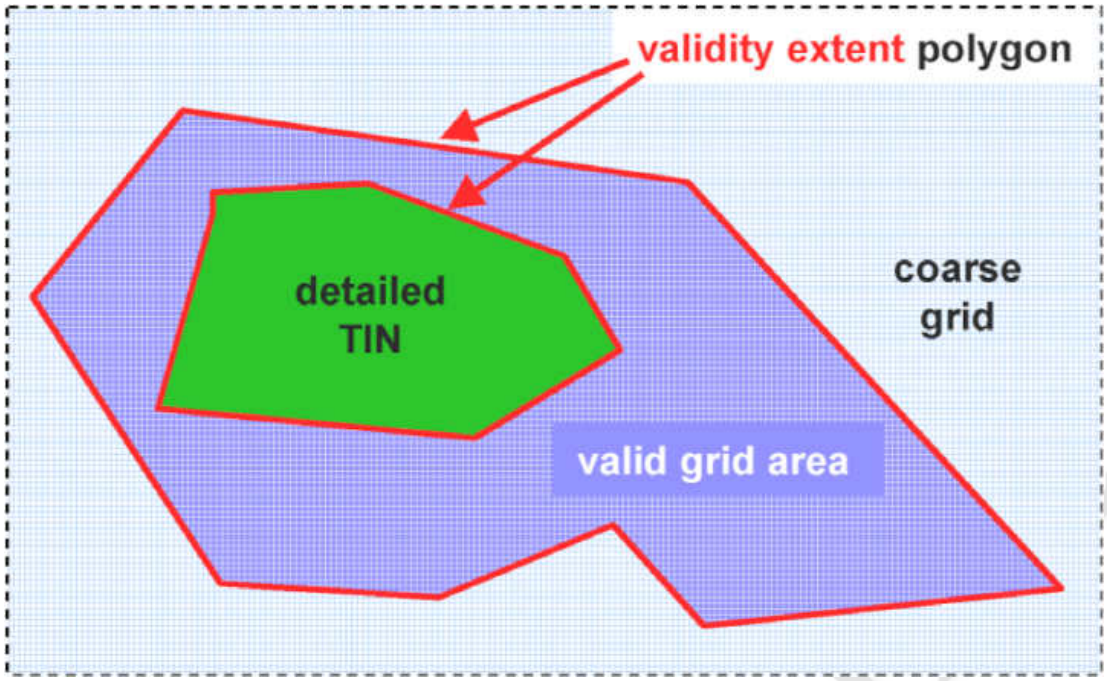

图 25. 使用有效范围多边形在CityGML中嵌套DTM（图：IGG Uni Bonn）。

DTM的精度和分辨率不一定依赖于其他CityGML扩展模块（如建筑模型）的特性。因此，有可能将LOD较高的建筑模型集成到精度或分辨率较低的DTM中。

这种方法与*TIC*相互作用（参见第6.5章）。TIC可以像断线一样用于调整DTM以适应不同的特征，如建筑物，桥梁或城市家具，从而确保DTM的一致表示。如有必要，可能需要处理重新成角。也可以通过DTM和相应特征的自交集求出TIC。

*ReliefFeature*和*ReliefComponent*都有一个*lod*属性，表示相应的细节级别。在大多数情况下，*ReliefFeature*的LOD与*ReliefComponent*的LOD匹配。但是，也允许指定具有高LOD的*ReliefFeature*，该*ReliefFeature*由*ReliefComponent*组成，其中一些*ReliefComponent*的LOD可以低于聚合*ReliefFeature*的LOD。其思想是，例如，对于LOD3场景，在LOD2中使用规则栅格以及由LOD3中*ReliefComponent*定义的某些高精度区域就足够了。LOD2网格和LOD3组件可以使用有效范围多边形的概念进行集成。因此，尽管一些*ReliefComponent*将被分类为较低的LOD，但是通过将其LOD值设置为3之后，*ReliefFeature*也可以与其他LOD3模型一起使用。

XML 命名空间

CityGML *Relief*模块的XML命名空间由统一资源标识符（URI）标识http://www.opengis.net/citygml/relief/2.0. 在*Relief*模块的xmlschema定义中，这个URI还用于标识默认名称空间。

10.2.1. 地形起伏特征和地形起伏组件

ReliefFeatureType, ReliefFeature

```
<xs:complexType name="ReliefFeatureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod" type="core:integerBetween0and4"/>
                <xs:element name="reliefComponent" type=
"ReliefComponentPropertyType" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfReliefFeature" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="ReliefFeature" type="ReliefFeatureType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfReliefFeature" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="ReliefComponentPropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="_ReliefComponent"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

AbstractReliefComponentType, _ReliefComponent

```
<xs:complexType name="AbstractReliefComponentType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod" type="core:integerBetween0and4"/>
                <xs:element name="extent" type="gml:PolygonPropertyType"
minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfReliefComponent" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="_ReliefComponent" type="AbstractReliefComponentType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfReliefComponent" type=
"xs:anyType" abstract="true"/>
```

10.2.2. 不规则三角网格地形

TINReliefType, TINRelief

```
<xs:complexType name="TINReliefType">
    <xs:complexContent>
        <xs:extension base="AbstractReliefComponentType">
            <xs:sequence>
                <xs:element name="tin" type="tinPropertyType"/>
                <xs:element ref="_GenericApplicationPropertyOfTinRelief"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="TINRelief" type="TINReliefType" substitutionGroup=
"_ReliefComponent"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTinRelief" type="
xs:anyType" abstract="true"/>
<!--
=======================================================================
============== -->
<xs:complexType name="tinPropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="gml:TriangulatedSurface"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

*TINRelief*的几何图形由GML几何图形类定义*gml:TriangulatedSurface*. 这部分可以显式地提供一组三角形(*gml:TriangulatedSurface*)或者通过使用子类（*gml:Tin* ）指定控制点、中断线和停止线。在后一种情况下，处理包含*gml:Tin* 需要通过使用约束Delaunay三角剖分算法重建三角剖分曲面（参见Okabe et al.1992）。

10.2.3. 栅格地形

RasterReliefType, RasterRelief, Elevation

```
<xs:complexType name="RasterReliefType">
    <xs:complexContent>
        <xs:extension base="AbstractReliefComponentType">
            <xs:sequence>
                <xs:element name="grid" type="gridPropertyType"/>
                <xs:element ref="
_GenericApplicationPropertyOfRasterRelief" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="RasterRelief" type="RasterReliefType" substitutionGroup
="_ReliefComponent"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRasterRelief" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== --><!--
================================================================================
============== -->
<xs:complexType name="gridPropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="gml:RectifiedGridCoverage"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="Elevation" type="gml:LengthType" substitutionGroup=
"gml:_Object"/>
```

10.2.4. 质量点地形

MassPointReliefType, MassPointRelief

```
<xs:complexType name="MassPointReliefType">
    <xs:complexContent>
        <xs:extension base="AbstractReliefComponentType">
            <xs:sequence>
                <xs:element name="reliefPoints" type=
"gml:MultiPointPropertyType"/>
                <xs:element ref=
"_GenericApplicationPropertyOfMassPointRelief" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="MassPointRelief" type="MassPointReliefType"
substitutionGroup="_ReliefComponent"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfMassPointRelief" type=
"xs:anyType" abstract="true"/>
```

10.2.5. 折断线地形

BreaklineReliefType, BreaklineRelief

```
<xs:complexType name="BreaklineReliefType">
    <xs:complexContent>
        <xs:extension base="AbstractReliefComponentType">
            <xs:sequence>
                <xs:element name="ridgeOrValleyLines" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="breaklines" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBreaklineRelief" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
==========================================================================
============== -->
<xs:element name="BreaklineRelief" type="BreaklineReliefType"
substitutionGroup="_ReliefComponent"/>
<!--
==========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBreaklineRelief" type=
"xs:anyType" abstract="true"/>
```

*BreaklineRelief*的几何图形可以由折断线和山脊/山谷线组成。折断线表示地形坡度的突变，脊线/谷线表示地形坡度符号的变化。*BreaklineRelief*必须至少包含上述两种类型中的一种。

10.2.6. 一致性要求

基础要求

1. *gml:Polygon*几何元素描述了使用 *_ReliefComponent*的范围属性（类型：__gml:PolygonPropertyType ）的有效范围，该元素应作为2D 底边轮廓多边形进行表示，该多边形可能具有内孔。

引用一致性

2. *ReliefFeature*元素的*reliefComponent*属性可以包含一个内嵌的 *_ReliefComponent*元素，或者直接使用GML 3.1.1的XLink概念，对远程 *_ReliefComponent*元素进行XLink引用。在后一种情况下 *_reliefComponent*属性中的*XLink:href*, 只能指向远程 *_ReliefComponent*元素（其中远程 *_ReliefComponent*元素位于另一个文档或同一文档中的其他位置）。另外，必须给出所包含的元素或引用，在这种条件下，只能选择同时给，或者同时不给。
3. *TINRelief*元素的*tin*属性可以包含一个内嵌的*gml:TriangulatedSurface*元素，或者 直接使用GML 3.1.1的XLink概念，对远程*gml:TriangulatedSurface* 元素进行XLink引用。在后一种情况下*TIN*属性中的*XLink:href* 属性，只能指向远程*gml:TriangulatedSurface*元素（其中远程*gml:TriangulatedSurface* 元素位于另一个文档或同一文档中的其他位置）。另外，必须给出所包含的元素或引用，在这种条件下，只能选择同时给，或者同时不给。

4. *RasterRelief*元素的*grid*属性可以包含一个内嵌的*gml:RectifiedGridCoverage* 元素，或者直接使用GML 3.1.1的XLink概念，对远程*gml:RectifiedGridCoverage*元素进行XLink引用。在后一种情况下*gml:RectifiedGridCoverage*元素的*XLink:href*只能指向远程*gml:RectifiedGridCoverage*元素（其中远程*gml:RectifiedGridCoverage*元素位于另一个文档或同一文档中的其他位置）。另外，必须给出所包含的元素或引用，在这种条件下，只能选择同时给，或者同时不给。

10.3. 建筑模型

建筑模型是CityGML最详细的主题概念之一。它允许在五个细节等级（LOD0到LOD4）中表示建筑物和建筑部件的主题属性和空间几何。CityGML的建筑模型由主题扩展模块*Building*定义（参见第7章）。图26提供了LOD1–4等级的3D城市和建筑模型示例。

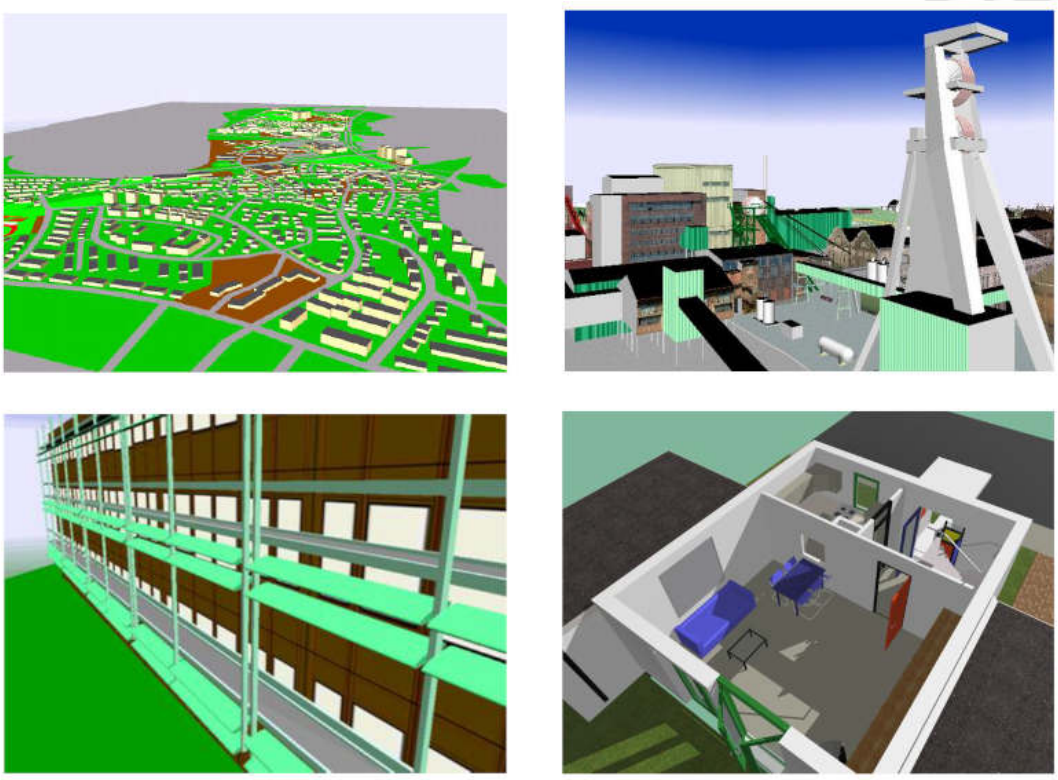

图 26. LOD1（左上），LOD2（右上），LOD3（左下）和LOD4（右下）中的城市或建筑模型示例（来源：雷克林豪森区，m-g-h ingenieure+architekten GmbH）。

建筑模型的UML图如图27所示，XML模式定义见附录A.4和下文。该模型的关键类是 *_AbstractBuilding*，它是主题类 *_Site*（继承自 *_CityObject*）的子类. *_AbstractBuilding*是专门针对建筑或建筑部件的。 因为 *_AbstractBuilding*是由建筑部件组成的，而建筑部件又可以是 *_AbstractBuildings*，所以可以实现任意深度的聚合层次结构。作为父类 *_CityObject*的子类， *_AbstractBuilding*继承来自 *_CityObject*的所有属性，就像GML3标准功能属性一样(*gml:name* etc.)以及CityGML特有的属性，如*外部引用*（参见第6.7章）。*_AbstractBuilding*未明确涵盖的其他属性可以通过CityGML *泛型*模块进行建模（参见第10.12章）或使用CityGML应用领域扩展机制（参见。 第10.13章）。

建筑综合体由许多不同的建筑组成，如工厂或医院综合体，应使用*CityObjectGroup*的概念进行聚合（参见第6.8章）。例如，综合体的主建筑可以表示为 “main building”。

*Building*和*BuildingPart*，都继承了 *_AbstractBuilding*的属性，如下：建筑物类别，功能（如住宅，公共或工业），用途，建造年份，拆除年份，屋顶类型，测量高度以及地上和地下楼层的数量和对应高度。这组参数适用

于粗略重建建筑物的三维形状，可由地籍系统提供。此外，对*Building*或*BuildingPart*也可以指定对应的*Address*特征。

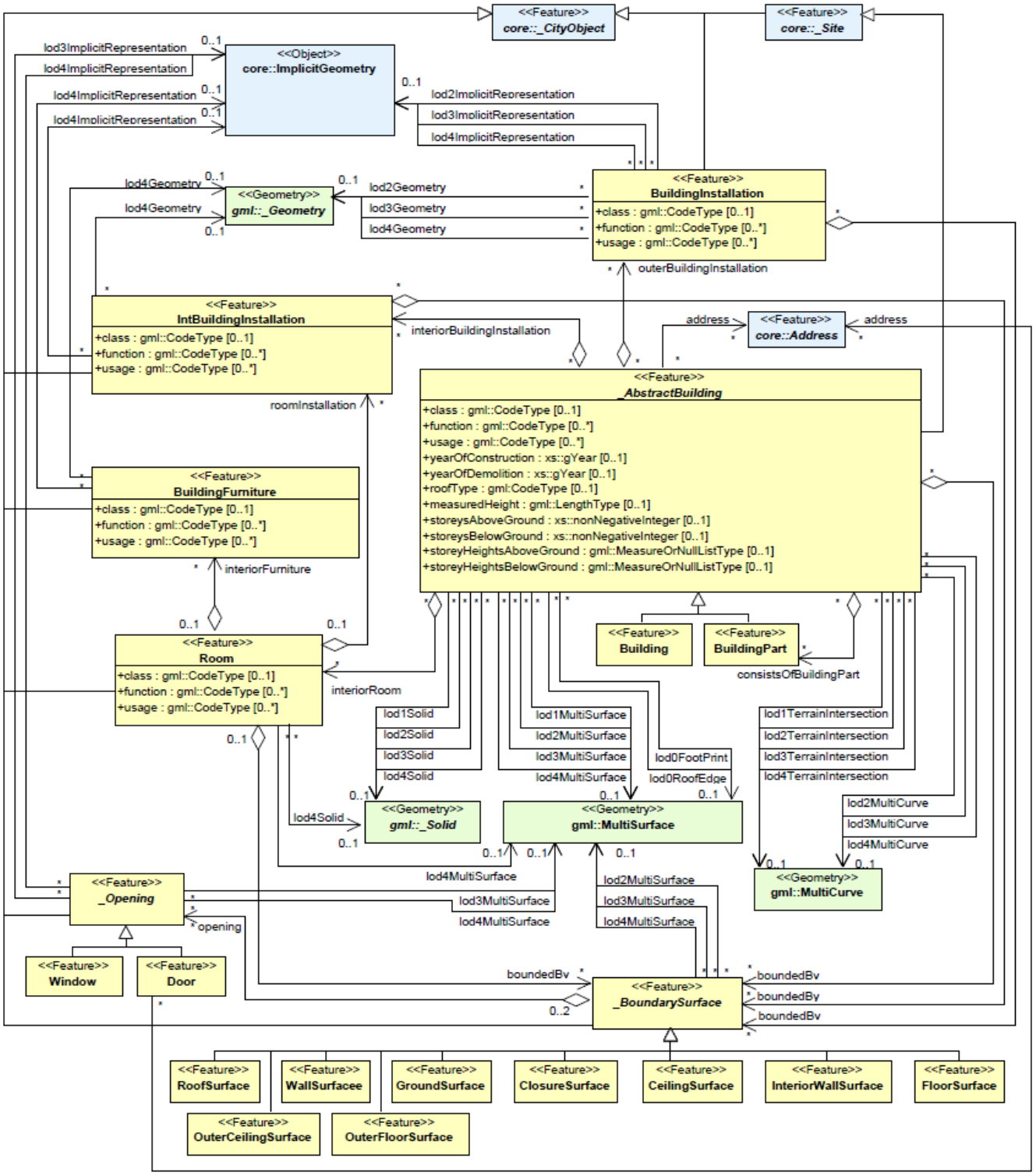

图 27. CityGML建筑模型的UML图。前缀用于指示与模型元素关联的XML名称空间。不带前缀的图元名称在CityGML Building模块中定义。

*_AbstractBuilding*的几何表示和语义结构如图27所示。模型从LOD0到LOD4依次细化。因此，并非建筑模型的所有部件在每个LOD中都具有等量的表示，并且并非每个LOD中都允许聚合。在CityGML中，所有对象类都与每个LOD的最小获取标准相关（参见第6.2章）。通过为相应的LOD提供不同的几何图形，可以在不同的LOD中表

示相同的对象。

在LOD0中，建筑可以用水平的三维表面来表示。这些可以分别代表建筑物的底边轮廓和屋顶边缘。这样可以轻松地将二维数据集成到模型中。在许多国家，这些二维几何图形很容易存在，例如地籍或地形数据。地籍数据通常描述地面上建筑物的底边轮廓，地形数据通常是底边轮廓和屋顶层（屋顶边缘）几何图形的混合体，通常从区域/卫星图像中通过摄影测量提取或从机载激光数据中提取。建筑模型允许包含上述两者。在这种情况下，可以将大型悬挑屋顶建模为更详细的LOD2和LOD3描述的初步阶段。曲面几何体需要三维坐标，但要求属于同一曲面的所有顶点，其高度值相同。如果将二维几何体导入这两个LOD0几何体中的任何一个，则需要为所有顶点选择适当的高度值。建筑物底边轮廓通常位于建筑物地面的最低标高处，而屋顶边缘表示应放置在屋顶标高处（例如屋檐高度）。

在LOD1中，建筑模型由建筑外壳的几何体表示。也可以选择 *gml:MultiCurve* 对*TIC*进行表示（参见第6.5章）。此几何表示在LOD2中通过附加*gml:MultiSurface* 和*gml:MultiCurve*，用于对屋顶悬挑，柱或天线等建筑细节进行建模。在LOD2和更高的LOD中，建筑物的外立面也可以通过 *_BoundarySurface* 和*BuildingInstallation*进行语义区分。 *_BoundarySurface*是建筑物外壳的一部分，具有墙（*WallSurface*），屋顶（*RoofSurface*），地面（*GroundSurface*），外部地面（*OuterFloorSurface*），外部天花板（*Outer-CeilingSurface*）或封闭面（*ClosureSurface*）等特殊功能。*BuildingInstallation*可用于建筑元素，如阳台，烟囱，老虎窗或外部楼梯，这些元素强烈影响建筑物的外观。一个*BuildingInstallation*可以具有属性*class*，*function*和*usage*（参见图27）。

在LOD3中， *_BoundarySurface*对象（门和窗）中的洞口可以表示为主题对象。在LOD4中(分辨率最高)，可以表达由多个房间组成的建筑内部，在建筑模型中由*Room*表示。这种可扩展性赋予了建筑物的虚拟信息承载性，例如博物馆中的游客信息（“基于位置的服务”），住宿标准的检查或建筑物日光照明的呈现。在该部分内容中，可通过使用CityGML提供的一般分组概念实现（参见第10.3.6章），基于用户定义的任意标准（例如，用于定义与某一楼层相对应的房间）对房间进行聚合。建筑物的内部装置，即建筑物内不能移动的物体（与家具不同），用*IntBuildingInstallation*表示。如果装置连接到特定房间（如散热器或灯具），则与房间类别相关，否则（如椽或管道）与建筑相关。一个*Room*可能具有属性*class*，*function*和*usage*，其值可在代码列表中定义（第10.3.8章和附录C.1）。*class*的属性允许根据所述功能对房间进行分类，例如商务或私人房间，并且该属性只能出现一次。*function*的属性可用于表示房间的主要用途，例如客厅，厨房。如果对象的实际使用方式与函数不同，则可以使用 *usage*属性。这两个属性可以出现多次。

房间的可见表面在几何上可以表示为*Solid*或*MultiSurface*。从语义上讲，曲面可以构造成特定的 *_BoundarySurfaces，表示地板*（FloorSurface）*，天花板*（CeilingSurface）*和内墙*（InteriorWallSurface）。*房间家具，如桌子和椅子，可以用*BuildingFurniture*表示。*BuildingFurniture*具有*class，funtion *和*usage__等属性。附录G.1至G.6提供了包含单个建筑模型的示例CityGML文件，该模型包含了从LOD0到LOD4模型，包括建筑内部。

XML 命名空间

CityGML *Building* 模块的XML命名空间由统一资源标识符（URI）标识http://www.opengis.net/citygml/building/2.0. 在*building*模块的xmlschema定义中，这个URI还用于标识默认名称空间。

10.3.1. 建筑和建筑部件

BuildingType, Building

```
<xs:complexType name="BuildingType">
    <xs:complexContent>
        <xs:extension base="AbstractBuildingType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfBuilding"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="Building" type="BuildingType" substitutionGroup=
"_AbstractBuilding"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBuilding" type="xs:anyType"
abstract="true"/>
```

*Building*是 *_AbstractBuilding*的两个子类之一。如果建筑物仅由一个（同质）部分组成，则应使用该类别。由不同结构段组成的建筑物，例如层数或屋顶类型不同的建筑物，必须被分割成一个具有一个或多个附加*BuildingPart*的*Building*（见图28）。建筑中心部分的几何和非空间特性应在聚合*Building*特征中表示。

BuildingPartType, BuildingPart

```
<xs:complexType name="BuildingPartType">
    <xs:complexContent>
        <xs:extension base="AbstractBuildingType">
            <xs:sequence>
                <xs:element ref="
_GenericApplicationPropertyOfBuildingPart" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="BuildingPart" type="BuildingPartType" substitutionGroup
="_AbstractBuilding"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBuildingPart" type=
"xs:anyType" abstract="true"/>
```

*BuildingPart*派生自 *_AbstractBuilding*。它用于对建筑物的部件进行建模（见图28）。*BuildingPart*对象应仅与一栋建筑或建筑部件相关。

图 28. 由一个建筑（右）和两个建筑部件（左）组成的建筑示例（来源：科堡市）。

AbstractBuildingType, _AbstractBuilding

```
<xs:complexType name="AbstractBuildingType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractSiteType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="yearOfConstruction" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="yearOfDemolition" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="roofType" type="gml:CodeType" minOccurs
="0"/>
                <xs:element name="measuredHeight" type="gml:LengthType"
minOccurs="0"/>
                <xs:element name="storeysAboveGround" type=
"xs:nonNegativeInteger" minOccurs="0"/>
                <xs:element name="storeysBelowGround" type=
"xs:nonNegativeInteger" minOccurs="0"/>
                <xs:element name="storeyHeightsAboveGround" type=
"gml:MeasureOrNullListType" minOccurs="0"/>
                <xs:element name="storeyHeightsBelowGround" type=
"gml:MeasureOrNullListType" minOccurs="0"/>
                <xs:element name="lod0FootPrint" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod0RoofEdge" type=
```

```
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="outerBuildingInstallation" type=
"BuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorBuildingInstallation" type=
"IntBuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded
"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="lod3Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="interiorRoom" type=
"InteriorRoomPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="consistsOfBuildingPart" type=
"BuildingPartPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="address" type="core:AddressPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfAbstractBuilding" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
```

```
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="_AbstractBuilding" type="AbstractBuildingType" abstract
="true" substitutionGroup="core:_Site"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAbstractBuilding" type=
"xs:anyType" abstract="true"/>
```

抽象类 *_AbstractBuilding*包含建筑属性，其几何表示以及建筑或建筑部件在不同LOD中的几何/语义表示。这些属性描述如下：

a) 建筑物或建筑部件（*class*），不同的预期用途（*function*）和不同的实际用途（*usage*）的分类。这些属性的数值可以在代码列表中指定。

b) 建造或建筑部件的建造年份（*yearOfConstruction*）和拆除年份（*yearOfDemolition*）。这些属性可用于描述城市模型中建筑发展的年表。时间点指的是真实世界的时间。

c) 建筑或建筑部分的屋顶类型（*roofType*）。可以在代码列表中指定此属性数值。

d) 建筑物或建筑物部分的测量相对高度（*measureHeight*）。

e) 地平面上（*storeyAboveGround*）和下（*storeyBelowGround*）的层数。

f) 地平面以上（*storeyHeightsAboveGround*）和以下（*storeyHeightsBelowGround*）的层高列表。列表中的第一个值表示最近楼层的高度。最后一个值是最远的高度。

在图29和图30中通过展示相同建筑物在5个LoD层级中的对应物，可以清楚阐明跨越不同等级的细节层次，建筑模型在地理度量表示的复杂度和粒度，以及将模型的主题结构划分为具有特殊语义含义的组件方面是不同的。在*_AbstractBuilding*里，有许多与某些LOD相关联的属性。

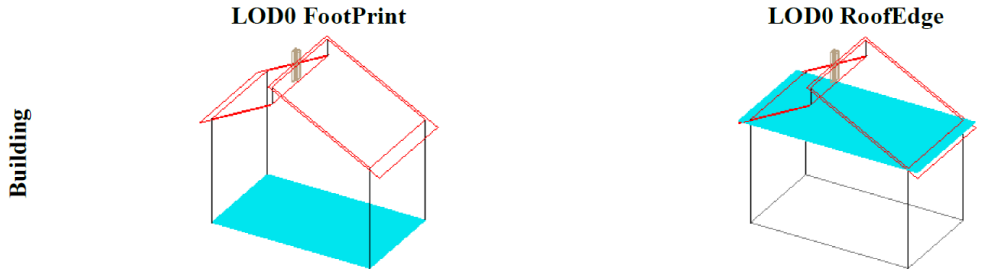

图 29. 使用水平三维曲面在LOD0中有两种建模方式。左侧显示建筑示意图（lod0FootPrint）（青色），表示地面上建筑的形状。相应的曲面表示位于地面。右图显示了lod0RooTEgde表示（青色），该表示来自建筑物屋顶的水平投影，位于屋檐高度（来源：KIT，由Franz提供）

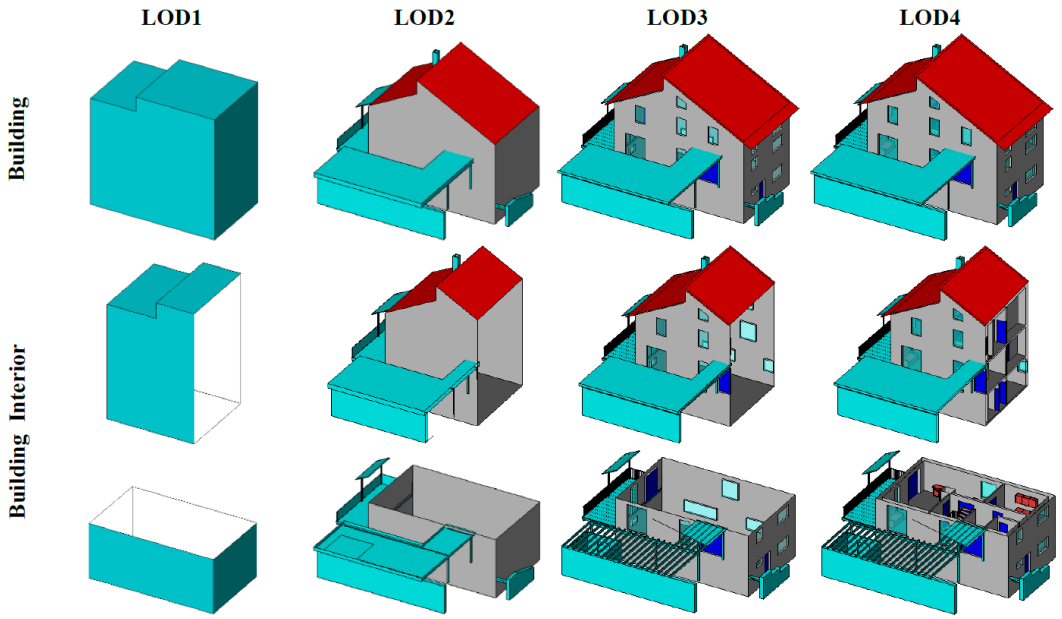

图 30. LOD1–LOD4中的建筑模型（来源：KIT，由Franz Josef Kaiser提供）

这五种类型，显示了建筑模型的不同几何和语义主题与LOD的对应关系。在LOD1–4中，建筑物的体积可以用*gml:Solid geometry*和/或*gml:MultiSurface geometry*来表示。三维地形相交曲线（TIC）的定义，用于将不同来源的建筑物与数字地形模型相结合，也可以在LOD1–4中定义。TIC（可选）在建筑物或建筑物部分周围构建闭合环。 在LOD0中（参见图29），建筑物由底边轮廓线和顶部水平表面表示。 在LOD1（参见图30）中，建筑物的不同结构实体被聚合到一个简单的块中，在细节上没有区别。建筑内外部不做区分，只使用一个相应的特性（*lod1Solid*或*lod1MultiSurface*）。

在LOD2和更高层次的细节中，建筑物的外部不仅在几何上表示为*gml:Solid geometry* 和/或*gml:MultiSurface geometry*，也可以由语义对象组成。在语义上构造建筑外部对象的基类，*_BoundarySurface*，该类型与*gml:MultiSurface* 有强联系（参见第10.3.2章）。 如果在建筑模型中既有利用体积/平面的建筑外部几何表示，又有通过 *_BoundarySurfaces*来进行语义定义的情况，则不能仅仅靠几何表示明确定义几何，必须引用 *_BoundarySurface* 中的*gml:MultiSurface*。

几何/语义主题	属性类型	LOD0	LOD1	LOD2	LOD3	LOD4
建筑底面轮廓与屋顶边界	*gml:MultiSurfaceType*	·				
建筑外壳体块部件	*gml:SolidType*		·	·	·	·
建筑外壳表皮部件	*gml:MultiSurfaceType*		·	·	·	·
地形相切曲线	*_gml:MultiCurveType*		·	·	·	·
建筑外壳曲线部件	*gml:MultiCurveType*			·	·	·
建筑部件	*BuildingPartType*		·	·	·	·

几何/语义主题	属性类型	LOD0	LOD1	LOD2	LOD3	LOD4
边界表面	*AbstractBoundarySurfaceType*			•	•	•
外部建筑装置	*BuildingInstallationType*			•	•	•
开口	*AbstractOpeningType*				•	•
房间	*RoomType*					•
内部建筑装置	*IntBuildingInstallationType*					•

除了*BuildingPart*外，建筑的较小部件（建构物/外部建筑装置）也会强烈影响建筑特征。这些特性由*BuildingInstallation*进行建模（参见第10.3.2章）。该类型中主要有，烟囱（参见。图30），老虎窗（见图28），阳台，外楼梯或天线。如果其范围超过第6.2章中规定的建议最小尺寸，则*BuildingInstallations*仅可包括在LOD2模型中。对于*BuildingInstallations*的几何表示，可以使用图9所示GML子集中的任意几何对象。

*_AbstractBuilding*没有可以使用为LOD3的属性。除了对几何精度要求较高和最小尺寸较小外，LOD2和LOD3建筑的主要区别在于 *_BoundarySurface*（参见第10.3.3章）。在LOD3中，通过使用抽象类 *_Opening*和其派生子类*Window*和*Door*，对建筑物中与窗或门相对应的开口（见图30）进行建模（参见第10.3.4章）。 对于建筑外部，LOD4数据模型与LOD3数据模型相同。但是LOD4提供了一种可能性，可以用*IntBuildingInstallation*和*Room*对建筑物的内部结构进行建模（参见第10.3.5章）。

可以使用*address*特性为每个*Building*或*BuildingPart*特征指定地址（数量自选）。相应的*AddressPropertyType*定义可以在CityGML核心模块中查看（参见第10.1.4章）。

10.3.2. 外部建筑装置

BuildingInstallationType, BuildingInstallation

```
<xs:complexType name="BuildingInstallationType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod2Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod3Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBuildingInstallation" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
============== -->
<xs:element name="BuildingInstallation" type="BuildingInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBuildingInstallation" type
="xs:anyType" abstract="true"/>
```

*BuildingInstallation*是建筑物的外部构件，它不具有建筑部件的意义，但对建筑物的外部特性有很大的影响。例如烟囱，楼梯，天线，阳台或楼梯和路径上方的附加屋顶。*BuildingInstallation*具有*class*, *function* 和 *usage*属性。属性类（只能出现一次）表示安装的一般分类。通过*function*和*usage*属性，可以描述建筑装置的名义功能和实际功能。对于上述三个属性，其数值可以在代码列表中指定。对于*BuildingInstallation*的几何表示，可以使用图9所示GML子集中的任意几何对象。或者，也可以将几何体作为*ImplicitGeometry*。根据*ImplicitGeometry*的概念，建筑装置的原型几何仅需要在局部坐标系中存储一次，在后续可以通过调取建筑

装置的特征进行引用（见第8.2章）。建筑物装置的可见表面可使用边界表面的概念进行语义分类（参见10.3.3）。*BuildingInstallation*只可以与一个建筑或建筑部件相关。

10.3.3. 边界表面

AbstractBoundarySurfaceType, *_BoundarySurface*

```
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="opening" type="OpeningPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBoundarySurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=========================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
=========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
"xs:anyType" abstract="true"/>
```

*_BoundarySurface*是几个主题类的抽象基类，用于构造建筑物的外表皮以及房间的可见表面以及外部和内部建筑装置。它是 *_CityObject*的一个子类，因此继承了所有属性，如GML3标准特征属性(*gml:name* etc.)以及CityGML特有的属性，比如*ExternalReferences*。从 *_BoundarySurface*出发，衍生出了屋顶表面（*RoofSurface*）, 墙面（*WallSurface*）, 地面（*GroundSurface*）, 外墙面 （*OuterCeilingSurface*），外地板表面（*OuterFloorSurface*）, 封闭表面（*ClosureSurface*），地板表面（*FloorSurface*）, 内墙面（*InteriorWallSurface*）和 天花板表面（*CeilingSurface*）等专题类。建筑表面的专题分类在图31（建筑外表皮）和图32（内表面）进行了说明。

对于Lod2到Lod4之间，几何体的 *_BoundarySurface*可以由不同的gml:MultiSurface geometry定义。

在LOD3和LOD4中， *_BoundarySurface*可能包含门和窗的*Openings*类（参见第10.3.4章）。如果开口的几何位置在拓扑结构上位于*gml:MultiSurface*几何体的曲面组件内（例如 *gml:Polygon*），这些开口必须表示为该曲

面内的孔。孔由相应曲面几何体对象内的内环表示。根据GML3，这些点必须以相反的顺序指定（当从曲面法向量的相反方向看时，外部边界为逆时针方向，内部边界为顺时针方向）。如果开口由*Door*, *Window*, 或*ClosureSurface*类型进行围合，则其外边界可由与周围表面的内环（表示孔）相同的点组成。*Openings*的边界表面属于相关的相邻边界表面。例如，如果门分隔开口，则门一侧属于*InteriorWallSurface*，另一侧属于*WallSurface*（图32右侧）。

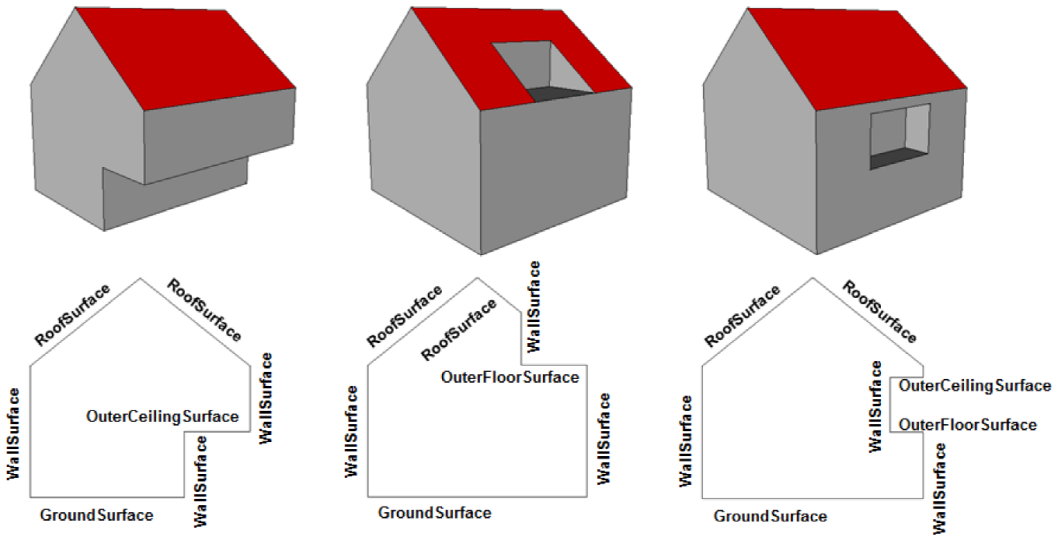

图 31. 建筑外壳的_BoundarySurfaces分类示例（来源：KIT）

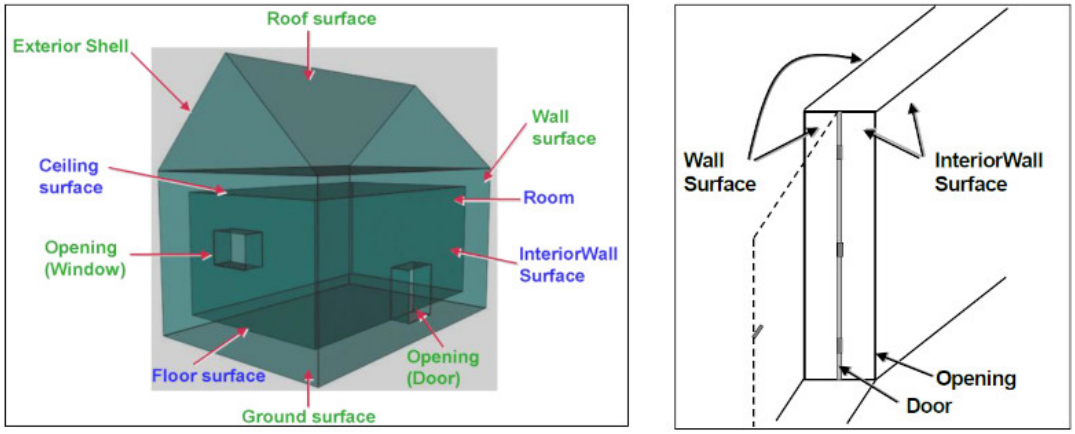

图 32. 边界表面分类（左），尤其是开口（右）（图：IGG Uni Bonn）。

GroundSurfaceType, GroundSurface

```
<xs:complexType name="GroundSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfGroundSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true"/>
```

通过*GroundSurface*，对建筑物或建筑部件的接地面进行建模。定义接地面的多边形与建筑物底面轮廓线一致。但是，接地板的表面法线向下。

OuterCeilingSurfaceType, OuterCeilingSurface

```
<xs:complexType name="OuterCeilingSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterCeilingSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true"/>
```

属于建筑外表皮且方向朝下的大部分水平表面可建模为*OuterCeilingSurface*。例如凉廊天花板或通道天花板的可见部分。

WallSurfaceType, WallSurface

```
<xs:complexType name="WallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWallSurface"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="WallSurface" type="WallSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWallSurface" type=
"xs:anyType" abstract="true"/>
```

建筑外立面的所有部分（该建筑外立面属于建筑外表皮的一部分）都可以通过类*WallSurface*进行建模。

OuterFloorSurfaceType, OuterFloorSurface

```
<xs:complexType name="OuterFloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
```

属于建筑外表皮且方向朝上的大部分水平表面可以建模为*OuterFloorSurface*。例如凉廊的地板。

RoofSurfaceType, RoofSurface

```
<xs:complexType name="RoofSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfRoofSurface"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true"/>
```

建筑物或建筑部件的主要屋顶部分可以用*RoofSurface*表示。具有特定语义的屋顶次要部分（如天窗或烟囱）应

建模为*BuildingInstallation*。

ClosureSurfaceType, ClosureSurface

```
<xs:complexType name="ClosureSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfClosureSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="true"/>
```

建筑物中没有门或窗的开口可以使用*ClosureSurface*（虚拟表面）进行分隔（参见第6.4章）。因此，像谷仓或机库这样的开放式建筑，为了能够计算它们的体积，可以考虑为关闭状态。*ClosureSurfaces*也用于内部建筑模型。如果两个具有不同功能的房间（如厨房和客厅）直接相连，而没有单独的门，则应使用*ClosureSurface*来分隔或连接两个房间的体积。

FloorSurfaceType, FloorSurface

```
<xs:complexType name="FloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="
_GenericApplicationPropertyOfFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
"xs:anyType" abstract="true"/>
```

*FloorSurface*只能在LOD4内部建筑模型中，用于对房间地板进行建模。

InteriorWallSurfaceType, InteriorWallSurface

```
<xs:complexType name="InteriorWallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfInteriorWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract="true"/>
```

InteriorWallSurface只能在LOD4内部建筑模型中使用，可对房间墙的可见表面进行建模。

CeilingSurfaceType, CeilingSurface

```
<xs:complexType name="CeilingSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfCeilingSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true"/>
```

*CeilingSurface*只能在LOD4内部建筑模型中使用，可对房间天花板进行建模。

10.3.4. 开口

AbstractOpeningType, _Opening

```
<xs:complexType name="AbstractOpeningType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfOpening"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
```

*_Opening*是一个抽象基类，用于在语义上描述墙和屋顶等外部或内部表面上的门或窗。开口（*_Opening*）仅存在于LOD3或LOD4模型中。每个开口都与类型为*gml:MultiSurface*的几何体进行关联。如果不关联的话，几何体需要作为其中的*ImplicitGeometry*。根据隐式几何的概念，开口的原型几何体仅需要在局部坐标系中存储一次，可由其他开口进行调用（见第8.2章）。

WindowType, Window

```
<xs:complexType name="WindowType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWindow"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
```

*Window*可用于对建筑外表皮中的窗口或相邻房间之间的舱口进行建模。*Window*和*Door*的形式上的区别在于，在正常情况下窗户并不是专门用来运送人或车辆的。

DoorType, Door

```
<xs:complexType name="DoorType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element name="address" type="core:AddressPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="_GenericApplicationPropertyOfDoor"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
```

*Door*可用于在建筑物的外表皮中或相邻房间之间，对门进行建模。人们可以用门进出建筑物或房间。

与*ClosureSurface*不同的是，门可能是关闭的，阻碍了人们的通行。一扇门可以分配多个地址。相应的*Address PropertyType*在CityGML核心模块中定义（参见第10.1.4章）。

10.3.5. 建筑内部

RoomType, Room

```
<xs:complexType name="RoomType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorFurniture" type=
"InteriorFurniturePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="roomInstallation" type=
"IntBuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded
"/>
                <xs:element ref="_GenericApplicationPropertyOfRoom"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="Room" type="RoomType" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRoom" type="xs:anyType"
abstract="true"/>
```

*Room*是一个语义对象，用于对建筑内部的自由空间进行建模，并且大部分情况下，只与一个建筑或建筑部件相关。*Room*为闭合的（如果需要的话，可以使用*ClosureSurface*），并且几何体通常用实体（*Lod4Solid*）来描述。然而，如果不能保证边界的拓扑正确性，几何体也可以作为一个*多曲面*（*Lod4MultiSurface*）给出。由于当作为GML外表皮时，其表面法线必须指向外部，因此当应*Room*被指定*Appearances*时，必须考虑这一点。在

这种情况下，纹理和颜色必须放置在相应曲面的背面，以便从房间内部可见。

除了几何表示之外，房间可见表面的不同部分可以通过特定的*BoundarySurfaces*（*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*，参见第10.3.3章）进行建模。

在为相邻房间之间的进行通道建模，需特殊考虑。房间实体通过用舱口，门或门口的封闭曲面进行拓扑连接。如果房间之间有共同的 *_Openings*或*ClosureSurfaces*，则房间被定义为相邻。在几何形体上表示开口的面是两个房间实体边界的一部分，或者该开口在语义级别上由两个房间引用。这种邻接意味着一种可达性图结构，可用于确定烟或气体的扩散，但也可用于使用经典最短路径算法计算逃生路线。

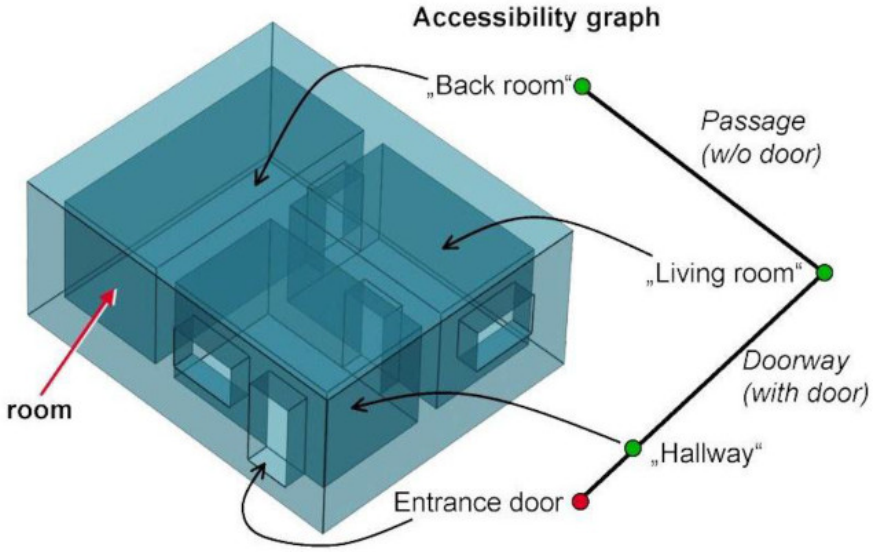

图 33. 从房间表面的拓扑邻接导出的可达性图（图：IGG Uni Bonn）。

BuildingFurnitureType, BuildingFurniture

```
<xs:complexType name="BuildingFurnitureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBuildingFurniture" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="BuildingFurniture" type="BuildingFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBuildingFurniture" type=
"xs:anyType" abstract="true"/>
```

房间里可能包含*BuildingFurnitures*和*IntBuildingInstallations*。*BuildingFurniture*是房间的可移动部分，如椅子或家具。*BuildingFurniture*对象应仅与一个房间对象相关。它的几何体可以由显式几何体或隐式几何体对象表示。根据*ImplicitGeometry*的概念，建筑家具的原型几何体仅在局部坐标系统中存储一次，可利用其特征进行调用（见第8.2章）。

IntBuildingInstallationType, IntBuildingInstallation

```
<xs:complexType name="IntBuildingInstallationType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfIntBuildingInstallation" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
============== -->
<xs:element name="IntBuildingInstallation" type=
"IntBuildingInstallationType" substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfIntBuildingInstallation"
type="xs:anyType" abstract="true"/>
```

*IntBuildingInstallation*是建筑物内具有特定功能或语义的对象。与*BuildingFurnitures*不同的是，*IntBuildingInstallations*永久附着在建筑结构上，不能移动。典型的例子是室内楼梯，栏杆，散热器或管道。*IntBuildingInstallation*类的对象既可以与*Room*相关联，也可以与完整的*Building/BuildingParth*相关联（*_AbstractBuilding*，参见第10.3.1章）。但是，它们应该仅与一个房间或一个建筑/建筑部分对象相关。*IntBuildingInstallation*具有*class*, *function* 和 *usage*属性。属性*class*只能出现一次，它表示内部建筑组件的一般分类。通过*function*和*usage*属性，可以描述建筑装置的名义功能和实际功能。对于上述所有三个属性，其数值可以在代码列表中指定。对于*IntBuildingInstallation*的几何表示，可以使用图9所示的GML子集中的任意几何对象。或者，几何体可以作为隐式几何体对象给出。根据隐式几何的概念，原型内部建筑安装的几何仅在本地坐标系统中存储一次，可基于其特征进行调用（见第8.2章）。内部建筑装置的可见表面可以使用边界表面的概念进行语义分类（参见10.3.3）。

10.3.6. 使用城市对象组对建筑楼层进行建模

由于AEC/FM标准IFC（IAI 2006）中提供了楼层表示，因此CityGML目前没有提供具体的概念。然而，一个楼层

可以用CityGML的城市对象组(*_CityObjectGroups*)概念表示为某一高度上所有建筑特征的显式集合（参见第10.11章）。这将包括*Rooms*，*Doors*，*Windows*，*IntBuilidingInstallations*和*BuildingFurniture*。如果主题表面（如墙和内墙）也应与特定楼层相关联，则可能需要对这些表面进行垂直分割（每层一个），因为在虚拟3D城市模型中，这些表面通常跨越整个建筑立面。

为了使用CityGML的通用分组概念对建筑楼层进行建模，必须使用 *CityObjectGroup*的嵌套层次结构。在第一步中，将属于特定层的所有语义对象分组。相应*CityObjectGroup*对象的属性设置如下：

- *class*属性应指定值“*建筑物分隔*”
- *function*属性应指定值“*lodX-Storey*”，X在1和4之间，以便注意该组的楼层数及特定的LOD等级。
- 楼层名称或编号可以存储在*gml:name*属性中. 楼层编号属性应用decimal X数值对“*storeyNo_X*”进行表示。

在第二步中，表示不同楼层的 *CityObjectGroup*将自己分组。通过使用 *CityObjectGroup*的通用聚合概念，“楼层组”与相应的*Building*或*BuildingPart*相关联。“*building storeys*”属性将被赋予至楼层*class*属性。

10.3.7. 示例

KIT北校区的LOD1模型如图34所示，由596栋建筑和187个建筑部分组成。建筑物底部轮廓线取自地籍信息系统，并按给定高度拉伸。具有唯一标识符和单个高度值的建筑建模为一个建筑(*bldg:Building*). 具有唯一标识符但高度值不同的建筑被建模为一个建筑(*bldg:Building*)一个或多个建筑部件(*bldg:BuildingPart*). 建筑和建筑部件都具有实体几何图形，它们的高度值另外表示为专题属性(*bldg:measuredHeight*). 图34显示了KIT Campus North（左）和CityGML LOD1模型（右）的航拍照片。

图 34. KIT北校区的LOD1模型（来源：KIT）。

全纹理LOD2建筑模型的示例如图35所示，该图显示了位于德国卡尔斯鲁厄市的Bernhardus教堂。在图35的左侧，示出了真实世界中教堂的照片，而右侧示出了具有照片级真实感纹理的教堂的CityGML建筑模型。该模型由一个接地面，多个墙曲面和屋顶曲面构成。教堂时钟上方的轨道线被建模为建筑装置（*BuildingInstallation*）。

图 35. 卡尔斯鲁厄伯恩哈德斯教堂的纹理LOD2模型（来源：KIT，卡尔斯鲁厄市提供）。

图36中所示的模型是从建筑物规划阶段生成的三维CAD模型导出的。在图36的左侧显示了建筑物，而在右侧显示了LOD3模型。建筑物本身由墙面，屋顶面和接地面构成。门和窗的模型包括分隔缝。根据地籍资料显示，大楼旁边的汽车端口不是大楼的一部分。因此，汽车港口，阳台和烟囱被建模为建筑设施（BuildingInstallation）。模型还包含建筑师规划的地形相交曲线（lod3TerrainIntersection）。

为了确定建筑的体积，建筑实体（lod3Solid）使用XLink机制引用了模型中所有边界曲面（包括门和窗）的几何图形。因此，屋顶曲面被拆分为表示屋顶本身的曲面和表示屋顶悬挑的曲面。

图 36. 在LOD3中建模的建筑示例。烟囱，阳台和汽车港口被建模为建筑安装（来源：KIT，由弗兰兹·约瑟夫·凯瑟提供）。

10.3.8. 代码列表

*_AbstractBuilding*的属性（*class*, *function*, *usage*和*roofType*），以及*BuildingInstallation*, *Room*, *BuildingFurniture* 和 *IntBuildingInstallation*的属性（*class*，*function*和*usage*）可以通过*gml:CodeType*进行规定. 这些属性的值可以在代码列表中枚举。有关相应代码清单的建议见附件C.1。

10.3.9. 一致性要求

基础要求

1. 如果*Building*仅由一个（同质）部件组成，则应以建筑部件表示。但是，如果建筑物由单独的结构件组成，则应将其建模为具有一个或多个*BuildingPart*的*Building*。只有建筑的几何和非空间属性才能在聚合建筑元素中表示。

基于不同lod的建筑模型构件使用限制

2. 当使用*gml:MultiSurface*几何形体关联上*LOD0FootPrint*和*LOD0RoofEdge*属性时，必须具有三维坐标。对于每个曲面，属于同一曲面的坐标元组，其高度值应相同。

3. *_AbstractBuilding*中*lodXSolid*，*lodXMultiSurface*属性，X∈[1..4]，（*gml:SolidPropertyType*，*gml:MultiSurfacePropertyType*）可用于表示每个LOD中建筑物的外表皮（作为体积或表面模型）。

4. 从LOD2开始，可以使用 *_AbstractBuilding*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）将*AbstractBuilding*的外表皮语义分解为*BoundarySurface*元素。其中，只允许将*RoofSurface*, *WallSurface*, *GroundSurface*, *OuterCeilingSurface*, *OuterFloor-Surface* 和 *ClosureSurface* 作为 *_BoundarySurface* 的子类。*boundedBy*属性（注意不要与*gml:*boundedBy__搞混)如果建筑物仅在LOD1中显示，则不得使用。

   当使用 *_BoundarySurface*元素，对建筑外表面进行表示时，需要有一个额外的形态对表面模型进行表示（使用*lodXSolid*和*lodXMultiSurface*, X∈[2..4]）。属性不应明确定义几何图形，但过程中必须使用XLink，对每层LOD中 *_BoundarySurface* 里*gml:MultiSurface* 元素中的相对应组件进行引用。

5. 从LOD2开始，建筑外表皮的曲线部分可以使用*lodXMultiCurve*， *_AbstractBuiilding*属性进行表示，建筑物的财产。但是如果建筑物仅在LOD1中表示，则不得使用该属性。

6. 从LOD2开始， *_AbstractBuilding*的*outerBuildingInstallation*属性（类型：BuildingInstallationPropertyType）可用于对BuildingInstallation元素建模。BuildingInstallation元素只能用于表示不具有建筑部件重要性的建筑外部特征。如果建筑物仅在LOD1中表示，则不得使用该属性。

7. 从LOD2开始，BuildingInstallation元素的几何体可以使用BuildingInstallation的*boundedBy*属性（类型：BoundarySurfacePropertyType），按 *_BoundarySurface*元素进行语义分类。其中，只允许将屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面作为 *_BoundarySurface*的子类。

8. 从LOD3开始，可以使用 *_BoundarySurface*的opening属性（类型：*OpeningPropertyType*）对*BoundarySurface*元素的开口进行建模。其中，当 *_BoundarySurface*元素仅在LOD2中表示时，不可以使用该属性。因此，在LOD2中表示 *_BoundarySurface*的曲面几何体必须拥有与其他曲面的连接。*_BoundarySurface*的opening属性可以包含或引用 *_Opening*元素。如果 *_Opening*元素的几何位置，在拓扑关系上位于边界曲面的曲面组件内，则开口也必须表示为该曲面的内孔。 *_Opening*元素的开口面，应属于对应的相邻边界表面，即 *_BoundarySurface*。

9. 从LOD4开始， *_AbstractBuilding*中的interiorRoom属性（类型： *InteriorRoomPropertyType*）可用于按*Room*元素对建筑内部的空间进行语义建模。如果建筑仅以LOD 1–3表示，则不得使用该属性。*Room*元素可以使用*lod4Solid / lod4MultiSurface*属性以几何形式，表示为体积或曲面模型(*gml:SolidPropertyType*，*gml:MultiSurfacePropertyType*)。 此外，房间可见表面的不同部分可以由主题边界表面(*_BoundarySurface*)元素建模。只允许将*FloorSurface*, *CeilingSurface*, *InteriorWallSurface* 和 *ClosureSurface*)作为 *_BoundarySurface*的子类。当房间的可见表面由 *_BoundarySurface*表示时，其中使用*lod4Solid*和*lod4MultiSurface*属性作为体积或表面模型的几何表示，不应明确定义几何体。但过程中必须使用XLink，对 *_BoundarySurface* 里*gml:MultiSurface*元素中的相对应组件进行引用。

10. 从LOD4开始， *_AbstractBuilding*的*interiorBuildingInstallation*属性（类型：*IntBuildingInstallationPropertyType*）可用于表示建筑物内永久附着在建筑结构上的不可移动对象。如果建筑物仅在LOD 1–3中表示，则不应使用*interiorBuildingInstallation*属性。此外，仅当对象无法与房间元素关联时，才应使用*interiorBuildingInstallation*属性。在后一种情况下，应使用相应房间元素的*roomInstallation*属性（类型：*IntBuildingInstallationPropertyType*）来表示对象。

11. 从LOD4开始，可以使用*IntBuildingInstallation*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）按*BoundarySurface*元素对*IntBuildingInstallation*元素的几何体进行语义分类。其中，只允许将*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*作为*_BoundarySurface*的子类。

引用一致性

12. *_AbstractBuilding*中的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）可以包含在内部定义过的 *_BoundarySurface*元素，或使用XLink对外部的 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href*属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

    只有*RoofSurface*，*WallSurface*，*GroundSurface*，*OuterCeilingSurface*，*OuterFloorSurface*和*ClosureSurface*元素，可以被 *_AbstractBuilding*的*boundedBy*属性进行封装或引用。

13. 元素 *_AbstractBuilding*的*outerBuildingInstallation*属性（类型：*BuildingInstallationPropertyType*）可以包含在内部定义过的*BuildingInstallation*元素或使用XLink对外部*BuildingInstallation*元素的进行引用。在后一种情况下 *_outerBuildingInstallation*属性的XLink:href 属性只能指向外部的 *_BuildingInstallation*元素（其中外部 *_BuildingInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

14. *_AbstractBuilding*元素的*interiorBuildingInstallation*属性（类型：*IntBuildingInstallationPropertyType*）可以包含在内部定义过的 *_IntBuildingInstallation*元素或使用XLink对外部 *_IntBuildingInstallation*元素进行引用。在后一种情况下 *interiorBuildingInstallation*属性的*xlink:href*属性只能指向外部的 *_IntBuildingInstallation*元素（其中外部 *_IntBuildingInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

15. *_AbstractBuilding*元素的*interiorRoom*属性（类型：*InteriorRoomPropertyType*）可以包含在内部定义过的 *_Room*元素或使用XLink对外部 *_Room*元素进行引用。在后一种情况下 *interiorRoom*属性的*xLink:href* 属性只能指向外部的 *_Room*元素（其中外部 *_Room*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

16. *_AbstractBuilding*的*consistsOfBuildingPart*属性（类型：*BuildingPartPropertyType*）可以包含在内部定义过的 *_BuildingPart*元素或使用XLink对外部 *_BuildingPart*元素进行引用。在后一种情况下*consistsOfBuildingPart*属性的*xlink:href*属性只能指向外部的 *_BuildingPart*元素（其中外部 *_BuildingPart*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

17. *_AbstractBuilding*的*address*属性（类型：*core:AddressPropertyTyp*)可以包含在内部定义过的*core:Address*元素或使用XLink对外部*core:Address*元素进行引用。在后一种情况下*address*属性的*xlink:href*属性只能指向外部的*core:Address*元素（其中外部*core:Address*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

18. *_BoundarySurface*的*opening*属性（类型：*OpeningPropertyType*）可以包含在内部定义过的 *_Opening*元素或使用XLink对外部 *_Openin*元素进行引用。在后一种情况下*opening*属性的*xlink:href*属性只能指向外部的 *_Opening*元素（其中外部 *_Opening*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

19. *Door*元素的地址属性可以包含在内部定义过的*core:Address*元素或使用XLink对外部*core:Address*元素进行引用。在后一种情况下*address*属性的*xlink:href* 属性只能指向外部的*core:Address*元素（其中外

部*core:Address*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

20. *BuildingInstallation*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

21. *IntBuildingInstallation*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 只有*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*元素，可以被*IntBuildingInstallation*的*boundedBy*属性进行封装或引用。

22. *Room*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 只有*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*元素，可以被*Room*的*boundedBy*属性进行封装或引用。

23. *Room*的*interiorFurniture*属性（类型：*InteriorFurniturePropertyType*）可以包含在内部定义过的*BuildingFurniture*元素或使用XLink对外部*BuildingFurniture*元素进行引用。在后一种情况下*interiorFurniture*属性的*xlink:href* 属性只能指向外部的*BuildingFurniture*元素（其中外部*BuildingFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

24. *Room*的*roomInstallation*属性（类型：*IntBuildingInstallationPropertyType*）可以包含在内部定义过的*IntBuildingInstallation*元素或使用XLink对外部*IntBuildingInstallation*元素进行引用。在后一种情况下*roomInstallation*属性的*xlink:href* 属性只能指向外部的*IntBuildingInstallation*元素（其中外部*IntBuildingInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

25. *BuildingInstallation*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

26. *BuildingInstallation*的*lod4ImplicitRepresentation*属性（类型：*core:ImplicitRepresentationPropertyType*）可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

27. *_Opening*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*)属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其

他位置）。该元素及引用，只能选择都给出，或都不给出。

28. *BuildingFurniture*的*lod4ImplicitRepresentation*属性（类型：*core:ImplicitRepresentationPropertyType*) 可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

10.4. 隧道模型

隧道模型与建筑模型密切相关。它支持在LOD1到LOD4四个级别中表示隧道和隧道部分的主题和几何信息。CityGML的隧道模型由主题扩展模块*Tunnel*定义（参见第7章）。图37提供了每个LOD的隧道模型的示例。

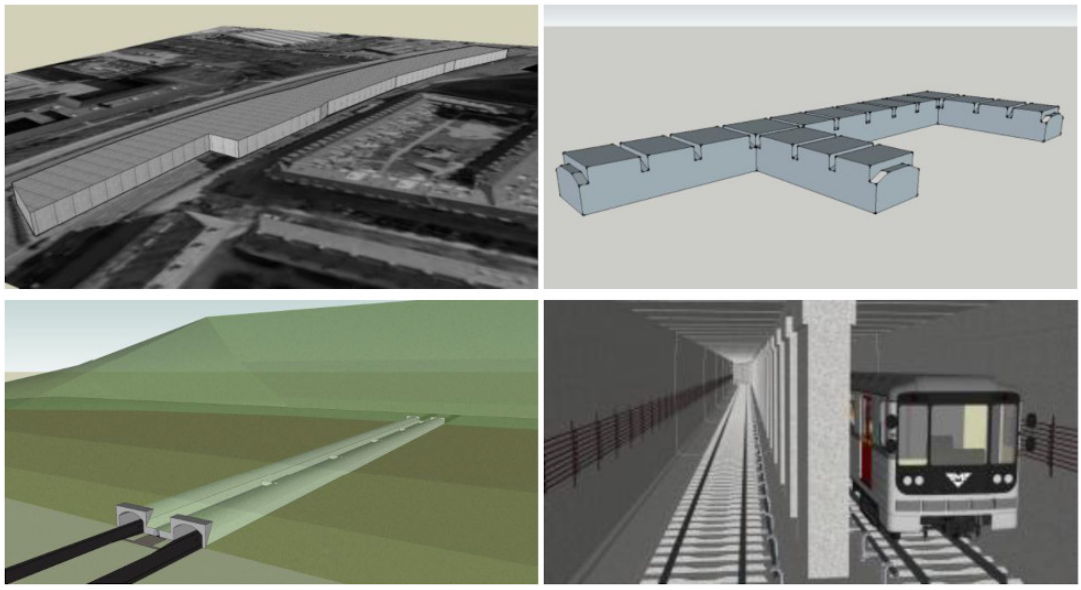

图 37. LOD1（左上），LOD2（右上），LOD3（左下）和LOD4（右下）中的隧道模型示例（来源：Google 3D warehouse）。

隧道模型的UML如图38所示。XML模式定义见附录A.11。该模型的关键类是 *_AbstractTunnel*，它是主题类 *_Site*（父类 *_CityObject*的衍生类）的子类。 *_AbstractTunnel*专用于*Tunnel*或*TunnelPart*。由于一个*_AbstractTunnel*由*TunnelParts*组成，而*TunnelParts*又是 *_AbstractTunnel*，因此可以实现任意深度的聚合层次结构。作为根类 *_CityObject*的子类， *_AbstractTunnel*继承 *_CityObject*的所有属性，如GML3标准特征属性(*gml:name* etc.)以及CityGML特有的属性，如外部引用（参见第6.7章）。 *_AbstractTunnel*未明确包含的其他属性可以建模为CityGML *泛型*模块（参见第10.12章）或使用CityGML应用领域扩展机制（参见第10.13章）提供的*泛型*属性。

*Tunnel*和*TunnelPart*两个类都继承了 *_AbstractTunnel*的属性：即，*Tunnel*的类，函数，用途，建设年份和拆除年份。与 *_AbstractBuilding*不同的是，*Address*特性不能分配给 *_AbstractTunnel*。

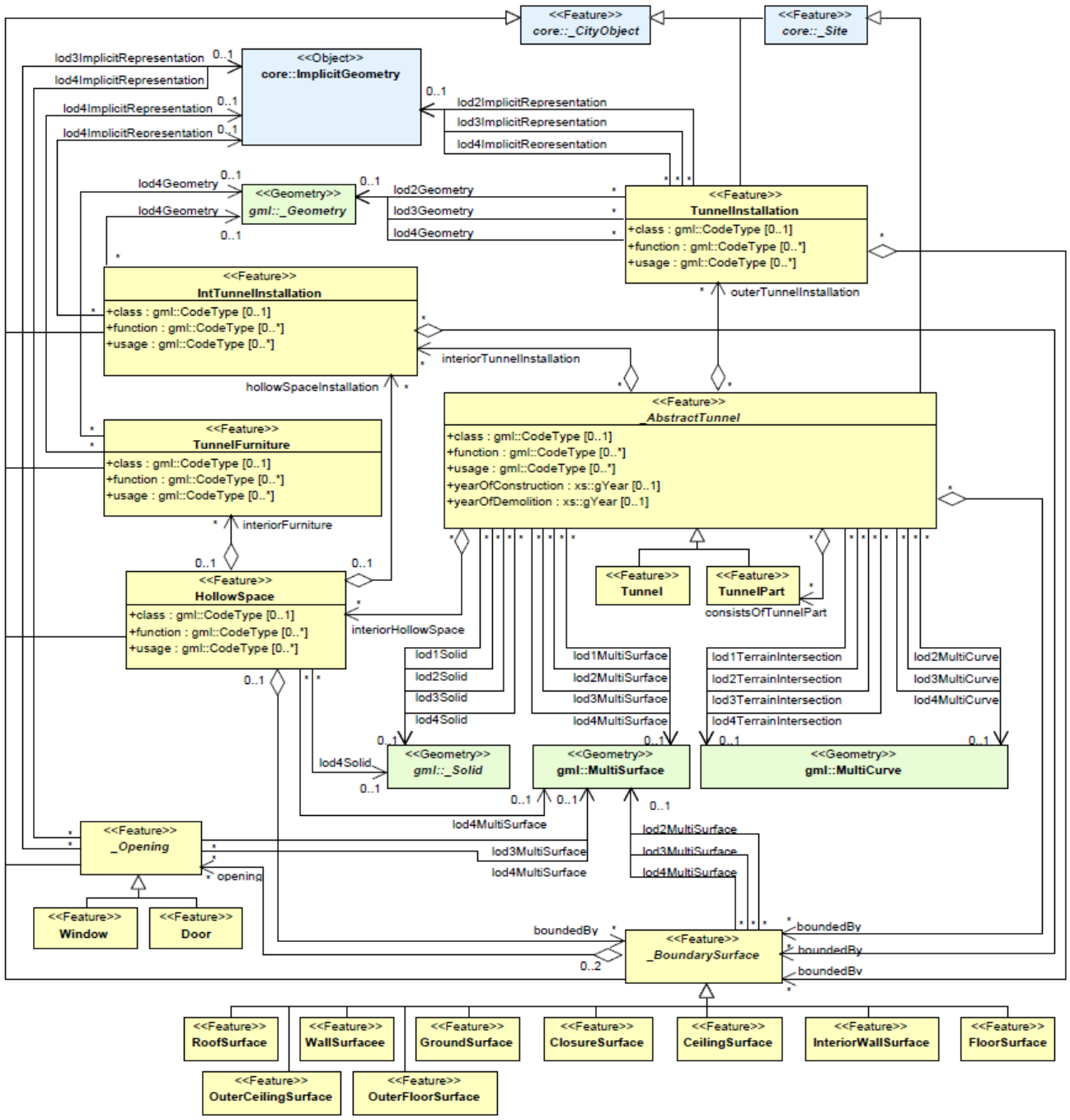

图 38. CityGML隧道模型的UML图。前缀用于指示与模型元素关联的XML名称空间。没有前缀的元素名称在CityGML隧道模块中定义。

*_AbstractTunnel*的几何表示和语义结构如图38所示。模型从LOD1到LOD4依次细化。因此，并非隧道模型的所有组件在每个LOD中都具有相同的表示，并且并非每个LOD中都允许进行聚合。在CityGML中，所有对象类都与每个LOD的建议最小标准相关（参见第6.2章）。通过为相应的LOD提供不同的几何图形，可以在不同的LOD中表示对象。

与建筑和桥梁模型类似（参见第10.3章和第10.5章），只有隧道的外壳在LOD1–3中表示，LOD1–3由隧道与周围土壤，水或室外空气的边界表面组成。隧道内部只能在LOD4中建模。尽管内部建筑环境与地下物体（如隧道或地下建筑）相关性较强，但CityGML对所有主题模块采用一致的LOD概念。相反，如果在所有LOD中都可以表示地下对象的内部，则地下对象的LOD概念必须与地上对象的LOD概念有很大的不同。这需要精确定义“过渡曲

面”，它界定了两个LOD概念的范围。此外，部分地上和地下的特征必须分为地上部分（根据地上LOD概念建模）和地下部分（根据地下LOD概念建模）。然而，这种分割违反了CityGML的特征统一概念，在许多情况下是不可行的，因为地上和地下之间的过渡，通常不能确切地知道或取决于地形模型的LOD。所以，CityGML将统一的LOD概念应用于地上和地下对象。结果如下，隧道与隧道内物体（如公路和铁路）之间的贯穿可能发生在LOD1–3。

在LOD1中，隧道模型由隧道空间的几何表示组成。或者，也可以指定一条表示地形剖面间曲线的多曲线（参见第6.5章）。在LOD2中，通过额外的多曲面和多曲线对几何表示进行了细化。

在LOD2和更高的LOD中，隧道的外部结构也可以通过 *_BoundarySurface*和*TunnelInstallation*进行语义区分。边界表面是隧道外壳的一部分，具有墙面（（*WallSurface*），屋顶表面（*RoofSurface*），地面（*GroundSurface*），外地板表面（*OuterFloorSurface*），外天花板（*OuterCeilingSurface*）或封闭表面（*ClosureSurface*）等特殊功能。*TunnelInstallation*主要用于会强烈影响隧道外观的外部楼梯，等类型的隧道元素。*TunnelInstallation*可以具有*class*，*function*和*usage*属性（参见图38）。

在LOD3中， *_BoundarySurface*对象（门和窗）中的开口可以表示为主题对象。

在LOD4中，最高级别的分辨率，也就是由多个空心空间组成的隧道内部，在隧道模型中由*HollowSpace*表示。上述的方式提高了隧道的虚拟可接近性，例如用于隧道行驶，用于模拟灾害管理或用于呈现隧道内的光照。通过采用CityGML提供的一般分组概念（参见第10.11章），根据用户定义的任意标准（例如，用于定义与水平或垂直截面相对应的空心空间）对空心空间进行聚合。隧道的内部装置，即隧道内不能移动的物体（与家具相反），由Int*TunnelInstallation*表示。如果装置连接到特定的中空空间（如灯具，通风机），则它们与HollowSpace相关联，否则（如管道）与*AbstractTunnel*相关联。一个HollowSpace可以具有*class*，*function*和*usage*属性，其可能的值可以在代码列表中枚举（第10.4.7章，附录C）。*class*属性允许对空心空间进行一般分类，例如商业或私人房间，并且只定义一次。*function*属性用于表示空心空间的主要用途，例如控制区域，安装空间，存储空间。如果对象的实际使用方式与实际功能不同，则可以使用*usage*属性。这两个属性可以出现多次。

空心空间的可见表面在几何上可以表示为*Solid*或*MultiSurface*。从语义上讲，曲面可以被构造成特定的*_BoundarySurfaces*，用来表示地板（*FloorSurface*），天花板（*CeilingSurface*）和内墙（*InteriorWallSurface*）。空心空间家具与控制区内的可移动设备一样，可以用具有*TunnelFurniture*属性的CityGML隧道模型来表示。*TunnelFurniture*可以具有*class*，*function*和*usage*等属性。

XML 命名空间

CityGML隧道模块的XML命名空间由统一资源标识符（URI）标识http://www.opengis.net/citygml/tunnel/2.0. 在*Tunnel*模块的XML模式定义中，此URI还用于标识默认名称空间。

10.4.1. 隧道和隧道部件

TunnelType, Tunnel

```
<xs:complexType name="TunnelType">
    <xs:complexContent>
        <xs:extension base="AbstractTunnelType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfTunnel"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="Tunnel" type="TunnelType" substitutionGroup=
"_AbstractTunnel"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnel" type="xs:anyType"
abstract="true"/>
```

*Tunnel*是 *_AbstractTunnel*的两个子类之一。如果隧道仅由一个（同源）部分组成，则应使用此类。由结构段组成的隧道，例如隧道入口和地铁，必须分为有一个或多个*TunnelPart*的隧道。隧道中心部分的几何特性和非空间特性应在聚合*Tunnel*中表示。

TunnelPartType, TunnelPart

```
<xs:complexType name="TunnelPartType">
    <xs:complexContent>
        <xs:extension base="AbstractTunnelType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfTunnelPart"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="TunnelPart" type="TunnelPartType" substitutionGroup=
"_AbstractTunnel"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnelPart" type=
"xs:anyType" abstract="true"/>
```

如果隧道的截面在几何形状和/或属性上不同，则可以将隧道分成多个部分（参见图39）。与*Tunnel*类似，*TunnelPart*派生自 *_AbstractTunnel*并继承 *_AbstractTunnel*的所有属性。*TunnelPart*对象应该只与一个隧道或隧道部件对象相关。

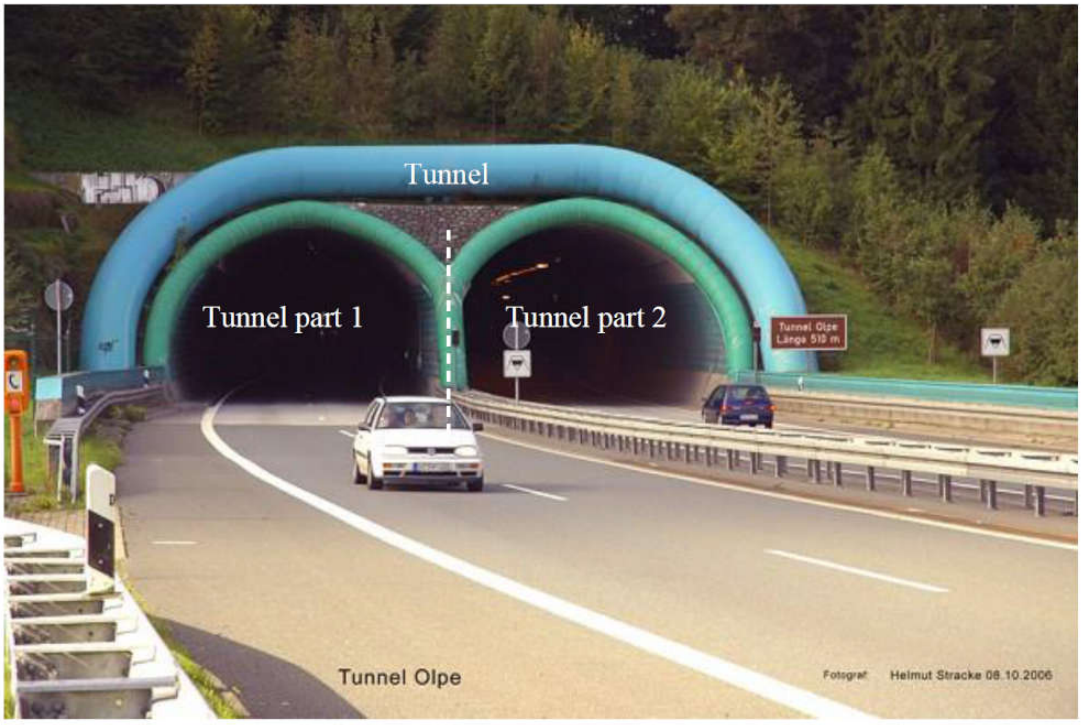

图 39. 使用两个隧道部件建模的隧道示例（来源：Helmut Stracke）。

AbstractTunnelType, *_AbstractTunnel*

```
<xs:complexType name="AbstractTunnelType">
    <xs:complexContent>
        <xs:extension base="core:AbstractSiteType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="yearOfConstruction" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="yearOfDemolition" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="lod1Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2Solid" type="gml:SolidPropertyType"
```

```
minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="outerTunnelInstallation" type=
"TunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorTunnelInstallation" type=
"IntTunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="lod3Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="interiorHollowSpace" type=
"InteriorHollowSpacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="consistsOfTunnelPart" type=
"TunnelPartPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfAbstractTunnel" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="_AbstractTunnel" type="AbstractTunnelType" abstract=
"true" substitutionGroup="core:_Site"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAbstractTunnel" type=
"xs:anyType" abstract="true"/>
```

抽象类 *_AbstractTunnel*包含隧道属性的属性，几何形体表示，以及隧道或隧道部分在不同细节级别的几何/语义表示。这些属性描述如下：

a) 隧道或隧道部分的分类（*class*），不同的功能（*function*）和用法（*usage*）。这些属性的类型是*gml:CodeType*，并且在单独的代码列表中对此赋值。

b) 隧道或隧道部分的建造年份（*yearOfConstruction*）和拆除年份（*yearOfDemolition*）。*yearOfConstruction*为隧道竣工的年份。*yearOfDemolition*是隧道拆除完成的年分。日期（年）指的是真实世界时间（如2011年）。

隧道模型跨越了不同的细节层次，其几何表示的复杂性和粒度不同，模型的主题结构也不同。这如图40所示，显示了四个不同LOD中的同一隧道。 *_AbstractTunnel*类，的某些属性也与lod级别相关。

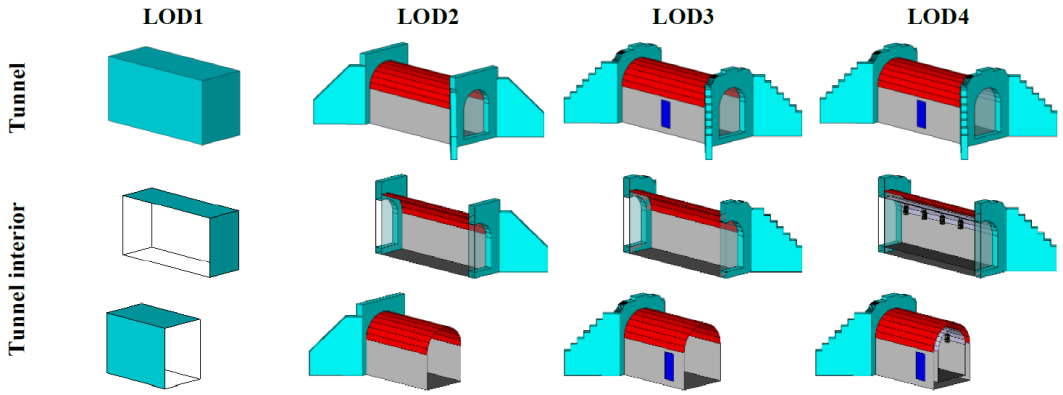

图 40. LOD1–LOD4中的隧道模型（来源：KIT）。

下方图表6，显示了隧道模型的几何表达和语义主题与LOD的对应关系。在每个LOD中，隧道的体积可以用*gml:Solid*几何 和/或*gml:MultiSurface* 几何. 三维地形相交曲线（TIC）的定义，可以用于将不同来源的隧道与数字地形模型相结合，也可用于所有LOD。TIC可以（但不是必须）在隧道或隧道部分周围建造封闭环。

几何/语义主题	属性类型	LOD1	LOD2	LOD3	LOD4
隧道外壳体块部件	*gml:SolidType*	·	·	·	·
隧道外壳表皮部件	*gml:MultiSurfaceType*	·	·	·	·
地形相切曲线	*_gml:MultiCurveType*	·	·	·	·
隧道外壳曲线部件	*gml:MultiCurveType*		·	·	·
隧道部件	*TunnelPartType*	·	·	·	·
边界表面	*AbstractBoundarySurfaceType*		·	·	·
外部隧道装置	*TunnelInstallationType*		·	·	·

几何/语义主题	属性类型	LOD1	LOD2	LOD3	LOD4
开口	*AbstractOpeningType*			·	·
中空空间	*HollowSpaceType*				·
内部建筑装置	*Int*TunnelInstallation*Type*				·

10.4.2. 外部隧道装置

TunnelInstallation*Type, *TunnelInstallation

*TunnelInstallation*是隧道的外部构件，它不具有*TunnelPart*的重要性，但对隧道的外部特性有很大影响，例如楼梯。*TunnelInstallation*具有*class*，*function*和*usage*属性。*class*类（只需定义一次）表示安装的一般分类。通过*function*和*usage*，可以描述隧道安装的名义和实际功能。对于所有三个属性，可在代码列表中进行指定。对于*TunnelInstallation*的几何表示，可以使用图9所示的GML子集中的任意几何对象。或者，其几何体可以作为*ImplicitGeometry*对象给出。根据*ImplicitGeometry*的概念，原型隧道安装的几何形体只需要在局部坐标系中存储一次，之后可以根据其特征被调用（见第8.2章）。隧道装置的可见表面可使用边界表面的概念进行语义分类（参见10.3.3）。*TunnelInstallation*对象应仅与一个*tunnel*或*tunnelPart*对象相关。

10.4.3. 边界表面

```
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="opening" type="OpeningPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBoundarySurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
"xs:anyType" abstract="true"/>
```

*_BoundarySurface*是多个主题类的抽象基类，用于构造隧道的外部外壳，空心空间的可见表面以及隧道的外部和内部装置。它是 *_CityObject*的一个子类，因此继承了所有属性，如GML3标准特征属性(*gml:name* etc.)以及CityGML特有的属性，比如*外部引用*。从边界表面出发，衍生出了如下专题类： *RoofSurface* , *WallSurface* , *GroundSurface* , *OuterCeilingSurface* , *OuterFloorSurface* , *ClosureSurface* , *FloorSurface* , *InteriorWallSurface* 和 *CeilingSurface*。对于不同类型的隧道横截面，隧道表面的专题分类如图41所示，具体如下。

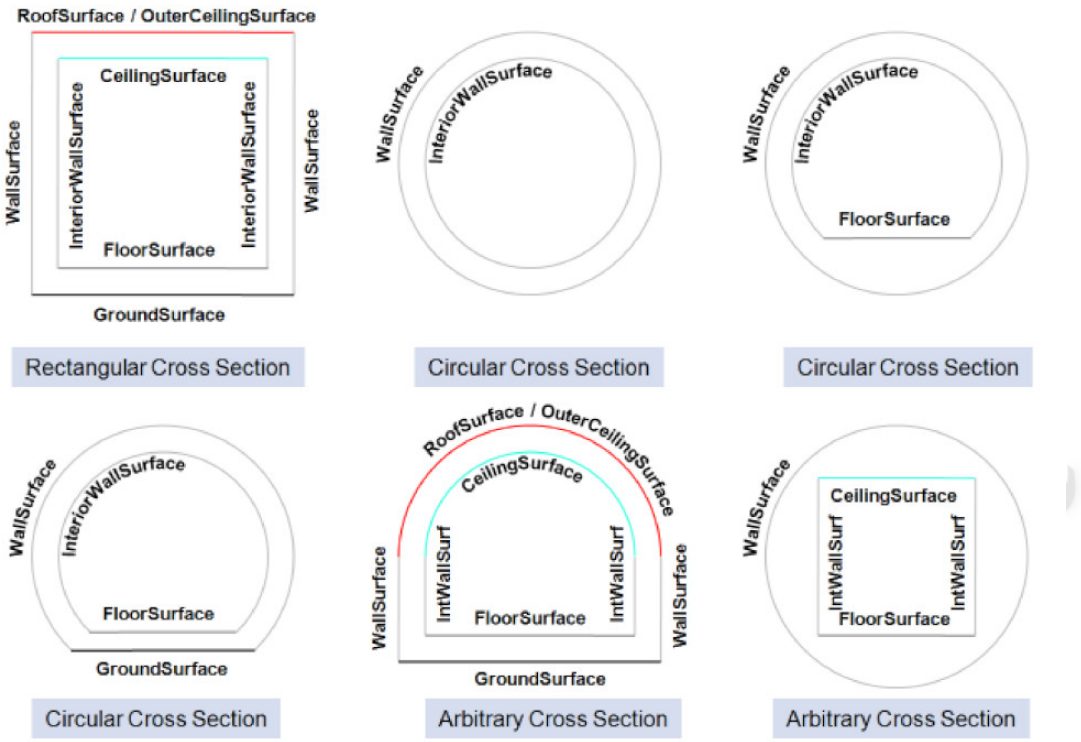

图 41. 不同横截面隧道的边界表面使用示例。墙表面，屋顶表面，地表面，外墙面和外地板表面在LOD2–4中可用，而内墙面，地板表面和天花板表面仅可在LOD4中用于建模空心墙的内部边界表面

对于2到4之间的LOD等级， *_BoundarySurface*的几何体可以由不同的*gml:MultiSurface* 几何进行定义. 从LOD3开始， *_BoundarySurface*可以包含门和窗之类的开口（参见第10.4.4章）。如果开口的几何位置在拓扑上位于*gml:MultiSurface*的曲面组件内（例如，*gml:Polygon*），这些开口需要表示为该曲面内的孔。孔由相应曲面几何体对象内的内环表示。根据GML3，点必须以相反的顺序指定（外部边界为逆时针方向，内部边界为顺时针方向，与曲面法向量方向相反）。如果此类开口由*Window*或*Door*密封，则其外边界可由与周围表面的内环（表示孔）相同的点组成。开口的边界表面属于相关的相邻边界表面。例如，如果一扇门封住了开口，那么门的一侧面属于*InteriorWallSurface*，另一侧的面属于*WallSurface*（参考图32的右部分了解建筑模型中的相同情况）。

GroundSurfaceType, GroundSurface

```
<xs:complexType name="GroundSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfGroundSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/> <!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true"/>
```

隧道或隧道部件的底板由*GroundSurface*进行建模。*GroundSurface*通常是隧道与周围土壤（土壤，岩石等）/水之间的边界表面。

OuterCeilingSurfaceType, OuterCeilingSurface

```
<xs:complexType name="OuterCeilingSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterCeilingSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true
```

属于隧道外表皮且方向朝下的大部分水平面可建模为*OuterCeilingSurface*。例如崩塌保护器的可见部分或隧道

与周围土壤/水之间的边界表面。

WallSurfaceType, WallSurface

```
<xs:complexType name="WallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWallSurface"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="WallSurface" type="WallSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfWallSurface" type=
"xs:anyType" abstract="true"/
```

属于隧道外壳的隧道立面的所有部分都可以通过*WallSurface*建模。通常，*WallSurface*是隧道与周围土壤（土壤，岩石等）或水之间的边界表面。

OuterFloorSurfaceType, OuterFloorSurface

```
<xs:complexType name="OuterFloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
```

属于隧道外表皮且方向朝上的大部分水平表面可建模为*OuterFloorSurfaceType*。

RoofSurfaceType, RoofSurface

```
<xs:complexType name="RoofSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfRoofSurface"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true
```

属于隧道外表皮并主要用于保护隧道不受上方影响的边界表面用*RoofSurface*表示。这些边界的方向主要是向上

的。

ClosureSurfaceType, ClosureSurface

```
<xs:complexType name="ClosureSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfClosureSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="true"/>
```

没有门或窗填充的隧道或中空空间中的开口，可以使用*ClosureSurface*密封（参见第6.4章）。例如，隧道的入口可以建模为*ClosureSurface*。

FloorSurfaceType, FloorSurface

```
<xs:complexType name="FloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="
_GenericApplicationPropertyOfFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
"xs:anyType" abstract="true"/>
```

*FloorSurface*只能用于LOD4内部隧道模型中，以便对空心空间的地板进行建模。

InteriorWallSurfaceType, InteriorWallSurface

```
<xs:complexType name="InteriorWallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfInteriorWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract=
```

*InteriorWallSurface*仅允许在LOD4内部隧道模型中用于建模空心空间的可见墙面。

CeilingSurfaceType, CeilingSurface

```
<xs:complexType name="CeilingSurfaceType">
    <xs:extension base="AbstractBoundarySurfaceType">
        <xs:sequence>
            <xs:element ref="_GenericApplicationPropertyOfCeilingSurface"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true"/>
```

*CeilingSurface*仅允许在LOD4内部隧道模型中使用，对空心空间的天花板进行建模。

10.4.4. 开口

AbstractOpeningType, _Opening

```
<xs:complexType name="AbstractOpeningType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfOpening"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
```

*_Openin*是一个抽象的基类，用于在语义上描述外边界和内边界上的门或窗等开口。开口仅存在于LOD3或LOD4模型中。每个开口都关联着*gml:MultiSurface*几何形体。或者，在使用开口的过程中，几何体可以作为*ImplicitGeometry*对象。根据*ImplicitGeometry*的概念，开口模型的原型几何仅需要在局部坐标系中存储一次，之后可通过开口模型的特征进行调用（见第8.2章）。

WindowType, Window

```
<xs:complexType name="WindowType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWindow"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
```

*Window*用于对隧道外壳和空心空间中的窗口或相邻空心空间之间的舱口进行建模。*Window*和*Door*的形式上的区别在于，在正常情况下窗户并不是专门用来运送人或车辆的。

DoorType, Door

```
<xs:complexType name="DoorType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfDoor"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
```

*Door*用于在隧道的外壳中或相邻空心空间之间对门进行建模。人们可以用门进出隧道或地下空间。与*ClosureSurface*不同的是，门可能会关闭，阻止人员或车辆通行。

10.4.5. 隧道内部

HollowSpaceType, HollowSpace

```
<xs:complexType name="HollowSpaceType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorFurniture" type=
"InteriorFurniturePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="hollowSpaceInstallation" type=
"IntTunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="_GenericApplicationPropertyOfHollowSpace"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
================ -->
<xs:element name="HollowSpace" type="HollowSpaceType" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfHollowSpace" type=
"xs:anyType" abstract="true"/>
```

*HollowSpace*是一个语义对象，用于对隧道内的自由空间进行建模，并且应该仅与一个隧道或隧道部分对象相关。该空间为闭合空间（可以使用*closurespace*），并且其几何体通常使用实体（*lod4Solid*）来描述。但是，如果不能保证边界的拓扑正确性，其几何体也可以是*MultiSurface*（*lod4MultiSurface*）。如果*HollowSpace*被指定为外观，则该几何的表面法线必须指向外部。在这种情况下，纹理和颜色必须放置在相应表面的背面，以便从中空空间内部可见。

除了几何表示之外，*HollowSpace*可见表面的不同部分可以通过特定的边界表面（*FloorSurface*, *CeilingSurface*, *InteriorWallSurface*, 和*ClosureSurface*，参见第10.4.3章）进行建模。

TunnelFurnitureType, TunnelFurniture

```
<xs:complexType name="TunnelFurnitureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfTunnelFurniture" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
============================================================================
=============== -->
<xs:element name="TunnelFurniture" type="TunnelFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfTunnelFurniture" type=
"xs:anyType" abstract="true
```

空心空间可能有*TunnelFurniture*。*TunnelFurniture*是中空空间的可移动部分。*TunnelFurniture*对象应仅与一个空心空间相关。它的几何体可以由显式几何体或隐式几何体对象表示。根据隐式几何的概念，*TunnelFurniture*的几何原型仅在局部坐标系中存储，可通过特征对该模型进行调用（见第8.2章）。

IntTunnelInstallationType, IntTunnelInstallation

```
<xs:complexType name="IntTunnelInstallationType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfIntTunnelInstallation" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="IntTunnelInstallation" type="IntTunnelInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfIntTunnelInstallation"
type="xs:anyType" abstrac
```

*IntTunnelInstallation*是隧道内具有特定功能或语义的对象。与*TunnelFurniture*不同，*IntTunnelInstallation*对象属于永久附着在隧道结构上，不能移动。典型的例子是室内楼梯，栏杆，散热器或管道。Int*TunnelInstallation*对象可以与中空空间（HollowSpace类）相关联，也可以与完整的隧道或隧道部分相关联（*IntTunnelInstallation*类，见第10.4.1章）。但是，它们应该仅与一个空心空间或一个隧道/隧道零件对象相关。

*IntTunnelInstallation*具有*class*, *function* 和 *usage*等属性。类属性只能需要定义一次，它表示内部隧道组件的分类。通过函数和用法，可以描述隧道安装的名义和实际功能。对于上述三个属性，可以在代码列表中对数值进行指定。对于*IntTunnelInstallation*的几何表示，可以使用图9所示的GML子集中任意几何对象。或者，几何体可以作为隐式几何体对象给出。根据隐式几何的概念，*IntTunnelInstallation*的几何原型仅需要在局部坐标系中存储一次，可通过其特征对其进行调用（见第8.2章）。*IntTunnelInstallation*的可见表面可使用边界表面的概念进行语义分类（参见10.4.3）。

10.4.6. 示例

图42所示为德国卡尔斯鲁厄市中心的人行地道。在图42的左侧，一张照片说明了真实世界的情况。地下通道的两个入口在照片中都用虚线矩形标出。在图的右侧，显示了CityGML隧道模型。为了使得整个隧道及其地下实体更清晰可见，隧道周围的地形从模型中移除。图43从不同的视角示出了相同的地下通道。相机位于左入口（图42中的黑色虚线矩形）的前面，并指向右入口的方向（图42中的白色虚线矩形）。在图43的右侧，从相同的透视图示出隧道模型。为了使隧道的地下部分可见，对地形表面进行开孔。模型的背景中显示了附近建筑物的LOD1表示。

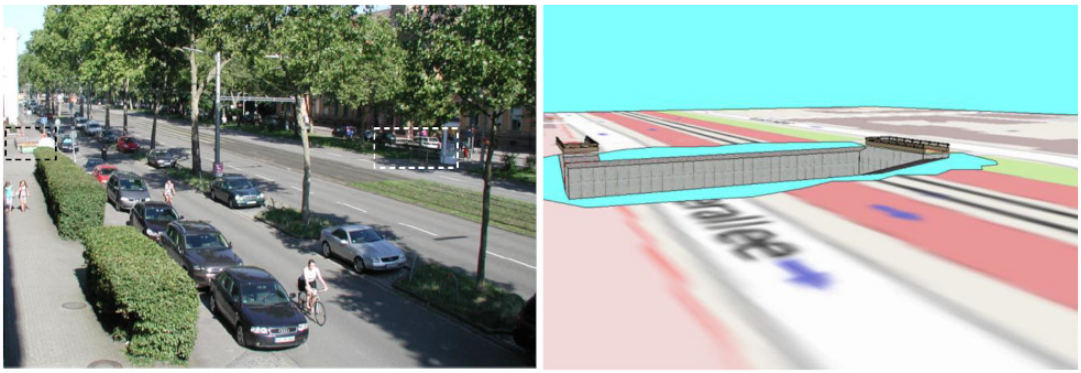

图 42. LOD3中建模的隧道示例（左侧为真实情况；右侧为CityGML模型）（来源：KIT，卡尔斯鲁厄市提供）。

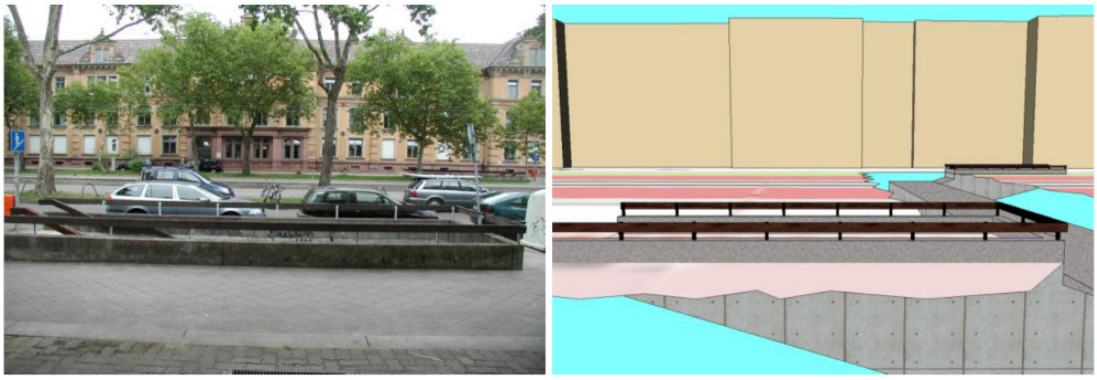

图 43. 从不同的角度显示相同的LOD3隧道。摄像机位于左入口前方，指向右入口方向。（左侧为真实情况；右侧为CityGML模型）。右侧的模型还包括背景中附近建筑物的LOD1表示（浅棕色绘制）（来源：KIT，由卡尔斯鲁厄市提供）。

上述模型细分为一个*Tunnel*（实际地下通道）和两个*TunnelPart*（两个入口）。隧道和隧道部件由*GroundSurface*, *WallSurface*, *RoofSurface*进行限定。*ClosureSurface*对象用于实际密封隧道入口。为安全起见，两个入口均设有栏杆，栏杆模型为*TunnelInstallation*。由于该模型具有较高的几何精度和语义丰富性，将其归类为LOD3。

10.4.7. 代码列表

AbstractTunnel，*TunnelInstallation*，*HollowSpace*，*TunnelFurniture*和*IntTunnelInstallation*的*class*, *function* 和 *usage*属性，指定为*gml:CodeType*。 这些属性的值可以在代码中通过枚举进行使用。相应代码见附录C.2。

10.4.8. 一致性要求

基础要求

1. 如果隧道仅由一个（同质）部分组成，则应使用*Tunnel*进行表示。但是，如果隧道由单个结构段组成，则应将其建模为具有一个或包含多个*TunnelPart*元素的*Tunnel*。只有隧道主体的几何和非空间特性才应在聚合*Tunnel*中表示。

基于不同LOD的隧道模型构件使用限制

2. *lodXSolid*和*lodXMultiSurface*，属性(*gml:SolidPropertyType* resp. *gml:MultiSurfacePropertyType*)在每个LOD中，隧道可以几何地表示隧道的外壳（作为体积或表面模型）。对于LOD1，必须使用*lod1Solid*或*lod1MultiSurface*，但不能同时使用两者。从LOD2开始，这两个属性可以单独使用。
3. 从LOD2开始， *_AbstractTunnel*的外表皮可以使用 *_AbstractTunnel*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）在语义上进一步分解为 *_BoundarySurface*元素。目前只允许将屋顶表面，墙面，地面，外墙面，外地板表面和封闭表面可以作为 *_BoundarySurface*的子类。*boundedBy*属性（注意：不要与*gml:boundedByproperty*混淆)如果隧道仅以LOD1表示，则不得使用。 当使用*_BoundarySurface*元素，对隧道外表面进行表示时，需要有一个额外的形态对表面模型进行表示（使用*lodXSolid*和*lodXMultiSurface*）。属性不应明确定义几何图形，但过程中必须使用XLink，对每层LOD中*_BoundarySurface* 里*gml:MultiSurface* 元素中的相对应组件进行引用。
4. 从LOD2开始，隧道外表皮可以使用*lodXMultiCurve*进行表示，属于 *_AbstractTunnel*。如果隧道仅在LOD1中表示，则不得使用该属性。
5. 从LOD2开始， *_AbstractTunnel*的*outerTunnelInstallation*属性（类型：*TunnelInstallationPropertyType*）可用于对*TunnelInstallation*元素建模。*TunnelInstallation*元素只能用于表示隧道的外部特征，而不具有隧道部件的重要性。如果隧道仅以LOD1表示，则不得使用*outerTunnelInstallation*属性。
6. 从LOD2开始，*TunnelInstallation*元素的几何体可以使用*TunnelInstallation*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）按*BoundarySurface*元素进行语义分类。目前只允许将屋顶表面，墙面，地面，外墙面，外地板表面和封闭表面作为 *_BoundarySurface*的子类。
7. 从LOD3开始，可以使用*BoundarySurface*的opening属性（类型：*OpeningPropertyType*）对*BoundarySurface*元素的开口进行建模。此属性不应用于仅在LOD2中表示的边界曲面元素。因此，在LOD2中表示边界曲面的曲面几何体属于全连接面。 *_BoundarySurface*的opening属性可以包含或引用*_Opening*元素。如果开口元素的几何位置在拓扑上位于边界曲面的曲面组件内，则开口也必须表示为该曲面的内孔。开口元素的边界表面应属于相邻边界表面。
8. 从LOD4开始， *_AbstractTunnel*的*interiorHollowSpace*属性（类型：*InteriorHollowSpacePropertyType*）可以通过*HollowSpace*元素对隧道内的空间进行语义建模。如果隧道仅在LOD 1–3中表示，则不应使用此属性。*HollowSpace*元素可使用*lod4Solid*或*lod4MultiSurface*属性，以几何形式表示为曲面或体积模型(*gml:SolidPropertyType* resp. *gml:MultiSurfacePropertyType*)。此外，空心空间可见表面的部件可由 *_BoundarySurface*元素进行建模。只允许将地板表面，天花板表面，内部墙面和封闭表面作为 *_BoundarySurface*的子类。如果空心空间的可见表面由 *_BoundarySurface*元素表示，则使用*lod4Solid*和*lod4MultiSurface*属性的体量/表皮模型不应明确定义几何图形，但过程中必须使用XLink，对 *_BoundarySurface* 里*gml:MultiSurface* 元素中的相对应组件进行引用。
9. 从LOD4开始， *_AbstractTunnel*的*interiorTunnelInstallation*属性（类型：Int*TunnelInstallationPropertyType*）可用于表示隧道内永久附着在隧道结构上的不可移动对象。如果

隧道仅在LOD 1–3中表示，则不应使用*interiorTunnelInstallation*属性。此外，仅当对象无法与*HollowSpace*元素关联时，才应使用*interiorTunnelInstallation*属性。当对象可以和*HollowSpace*进行关联时，应使用相应*HollowSpace*元素的*hollowspaceinstallation*属性（类型：*IntTunnelInstallationPropertyType*）来表示对象。

10. 从LOD4开始，*IntTunnelInstallation*元素的几何体可以使用*IntTunnelInstallation*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）按*BoundarySurface*元素进行语义分类。目前只允许将*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*作为 *_BoundarySurface*的子类。

引用一致性

11. *_AbstractTunnel*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

    只有屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面元素才允许由 *_AbstractTunnel* 的*boundedBy*属性进行封装或引用。

12. *_AbstractTunnel*的*outerTunnelInstallation*（类型：*TunnelInstallationPropertyType*）属性可以包含在内部定义过的*TunnelInstallation*元素或使用XLink对外部*TunnelInstallation*元素进行引用。在后一种情况下*outerTunnelInstallation*属性的*xlink:href* 属性只能指向外部的*TunnelInstallation*元素（其中外部*TunnelInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

13. *_AbstractTunnel*的*interiorTunnelInstallation*（类型：*IntTunnelInstallationPropertyType*）属性可以包含在内部定义过的*IntTunnelInstallation*元素或使用XLink对外部*IntTunnelInstallation*元素进行引用。在后一种情况下*interiorTunnelInstallation*属性的*xlink:href* 属性只能指向外部的*IntTunnelInstallation* 元素（其中外部*IntTunnelInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

14. *_AbstractTunnel*的*interiorHollowSpace*（类型：*InteriorHollowSpacePropertyType*）属性可以包含在内部定义过的*HollowSpace*元素或使用XLink对外部*HollowSpace*元素进行引用。在后一种情况下*interiorHollowSpace*属性的*xlink:href* 属性只能指向外部的*HollowSpace*元素（其中外部*HollowSpace*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

15. *_AbstractTunnel*的*consistsOfTunnelPart*（类型：*TunnelPartPropertyType*）属性可以包含在内部定义过的*TunnelPart*元素或使用XLink对外部*TunnelPart*元素进行引用。在后一种情况下*consistsOfTunnelPart*属性的*xlink:href* 属性只能指向外部的*TunnelPart*元素（其中外部*TunnelPart* 元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

16. *_BoundarySurface*的*opening*（类型：*OpeningPropertyType*）属性可以包含在内部定义过的 *_Opening* 元素或使用XLink对外部 *_Opening*元素进行引用。在后一种情况下opening属性的*xlink:href* 属性只能指向外部的 *_Opening*元素（其中外部 *_Opening*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

17. *TunnelInstallation*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况

下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部*_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

18. *IntTunnelInstallation*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部*_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

    只有*FloorSurface*, *CeilingSurface*, *InteriorWallSurface* 和 *ClosureSurface*元素才允许由*_AbstractTunnel*的*boundedBy*属性进行封装或引用。

19. *HollowSpace*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的*_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部*_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 只有屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面元素才允许由*HollowSpace*的*boundedBy*属性进行封装或引用。

20. *HollowSpace*的*interiorFurniture*（类型：*InteriorFurniturePropertyType*）属性可以包含在内部定义过的*TunnelFurniture*元素或使用XLink对外部*TunnelFurniture*元素进行引用。在后一种情况下*interiorFurniture*属性的*xlink:href* 属性只能指向外部的TunnelFurniture元素（其中外部*TunnelFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

21. *HollowSpace*的*hollowSpaceInstallation*（类型：*IntTunnelInstallationPropertyType*）属性可以包含在内部定义过的*IntTunnelInstallation*元素或使用XLink对外部*IntTunnelInstallation*元素进行引用。在后一种情况下*hollowSpaceInstallation*属性的*xlink:href* 属性只能指向外部的*IntTunnelInstallation*元素（其中外部Int*TunnelInstallation*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

22. *TunnelInstallation*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

23. *IntTunnelInstallation*的*lod4ImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

24. _Opening的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

25. *TunnelFurniture*的*lod4ImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

10.5. 桥梁模型

桥梁模型允许在四个细节层次（LOD 1–4）中表示桥梁和桥梁部分的主题，空间和视觉方面。CityGML的桥梁模型由主题扩展模块bridge定义（参见第7章）。图44示出了所有lod中的桥模型的示例。

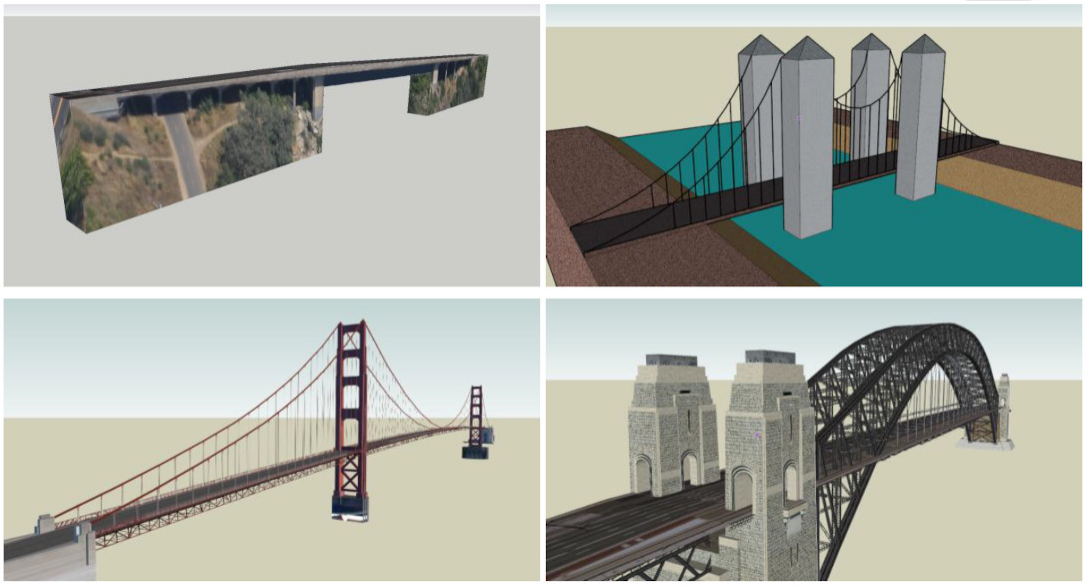

图 44. LOD1（左上），LOD2（右上），LOD3（左下）和LOD4（右下）中的桥梁模型示例（来源：Google 3D warehouse）

桥梁模型在结构和属性方面类似于建筑模型（参见第10.3章）。桥梁模型的UML图如图45所示，XML模式定义如附录A.3所示。

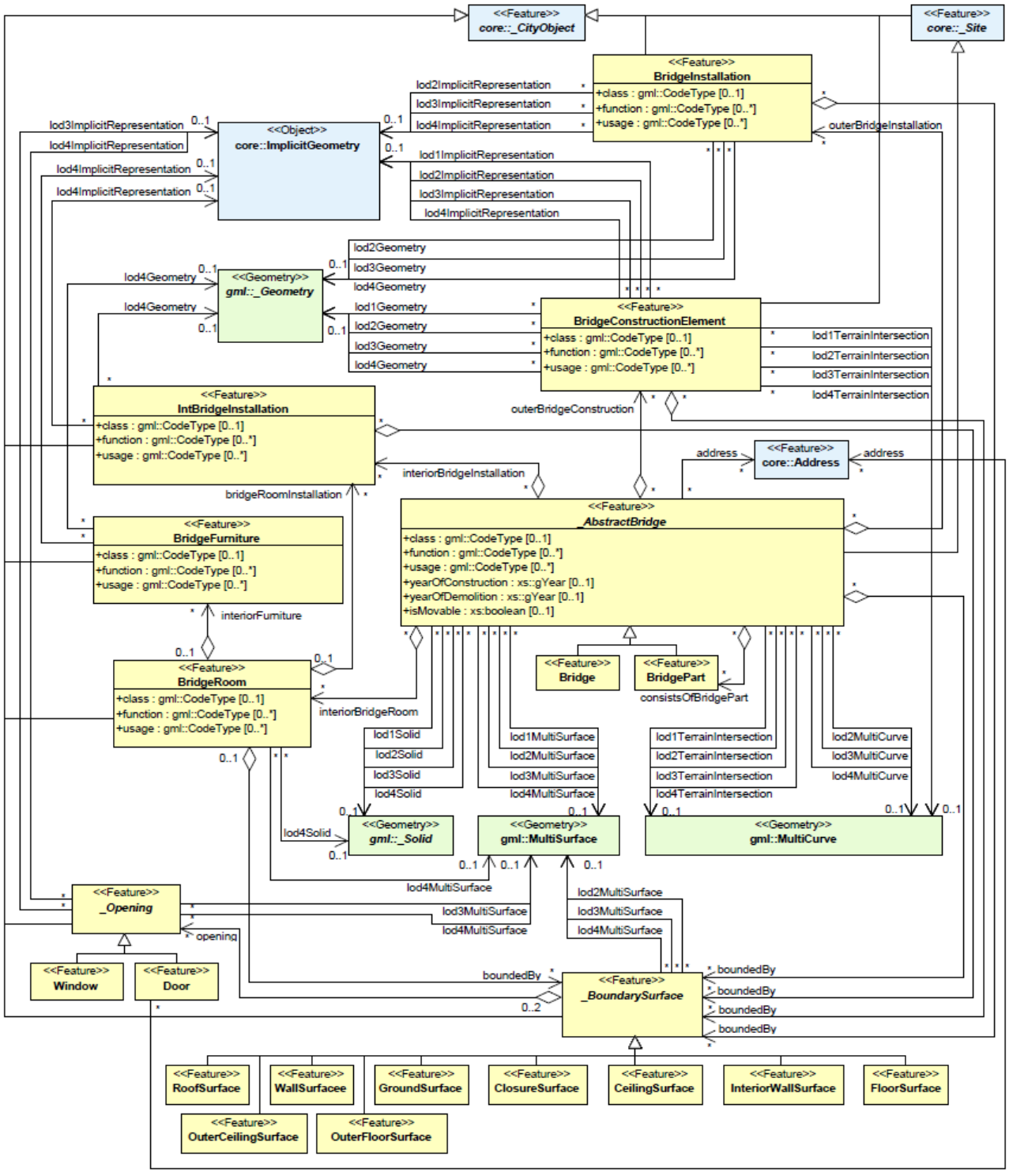

图 45. 桥梁模型的UML图，第一部分。

（可移动或不可移动的）桥由Bridge类的对象表示。这个类从抽象基类 *_AbstractBridge*衍生，并继承其属性和关系。空间特性由四个LOD（*lod1Solid*到*lod4Solid*的关系）中的每一个的实体定义。与建筑模型类似，从LOD1（块模型）到LOD3（建筑模型），语义和几何的丰富性都增加了。每个LOD中桥梁的简单示例如图46所示。室内结构（如房间）专用于LOD4。为了涵盖拓扑不满足实体特性（本质上是水密性）的桥梁模型情况，允许使用多曲面表示（*lod1MultiSurface*到*lod4MultiSurface*）。桥梁与地形表面接触的线由地形交点曲线表示。除了桥梁的实体表示外，绳索或天线等线性特征还可以通过LOD1多曲线到LOD4多曲线关系进行几何指定。如果这些特征应以语义表示，则可使用特征*BridgeFurniture*或*BridgeConstructionElement*（见第10.5.2节）。所

有与语义对象和地理度量属性的关系都列在下图表7中。

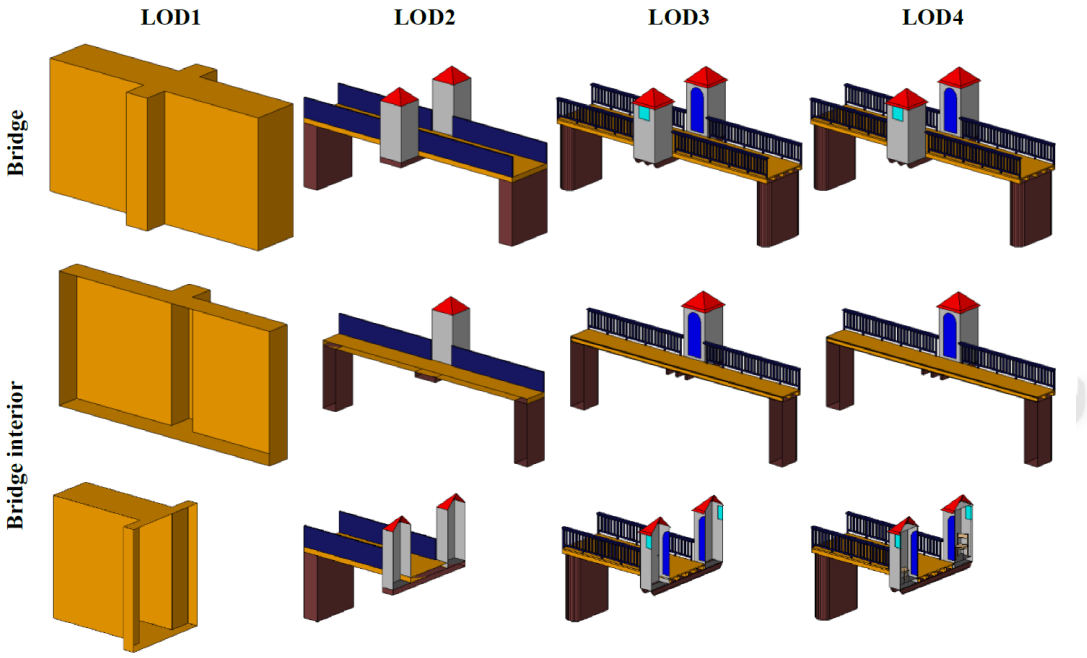

图 46. 桥梁模型的UML图，第一部分。

*_AbstractBridge*的语义属性是*class*，*function*，*usage*和_movable。*class*属性用于对桥梁进行分类，例如区分不同的结构类型（参见图48）。*function*属性允许独立于构造表示桥梁的利用率。其可选用的值可以是铁路桥，公路桥，人行天桥，渡槽等。当桥梁的功能，与上述功能不同时，可以使用属性*usage*进行赋值。这些属性的类型是*gml:CodeType*，其值可以在代码列表中定义。桥的名称可以用*gml:name* 属性，它通过*gml:_Feature*，*_CityObject*和 *_Site*从基类*gml:_gml*继承。可以使用*address*属性为每个桥或桥零件特征分配零个或多个地址。相应的*AddressPropertyType*在CityGML核心模块中定义（参见第10.1.4章）。

几何/语义主题	属性类型	LOD1	LOD2	LOD3	LOD4
桥梁外壳体块部件	*gml:SolidType*	·	·	·	·
桥梁外壳表皮部件	*gml:MultiSurfaceType*	·	·	·	·
地形相切曲线	*_gml:MultiCurveType*	·	·	·	·
桥梁外壳曲线部件	*gml:MultiCurveType*		·	·	·
桥梁部件	*BridgePartType*	·	·	·	·
边界表面	*AbstractBoundarySurfaceType*		·	·	·
外部桥梁装置	*BridgeFurnitureType*		·	·	·

几何/语义主题	属性类型	LOD1	LOD2	LOD3	LOD4
桥梁建造要素	*BridgeConstructionElementType*	·	·	·	·
开口	*AbstractOpeningType*			·	·
桥梁空间	*_BridgeRoomType_*				·
内部建筑装置	*IntBridgeFurnitureType*				·

boolean属性*is_movable*，用于指定桥是否可移动。可运动的几何建模延迟到本标准的后续版本。图47中描绘了一些类型的可移动桥。

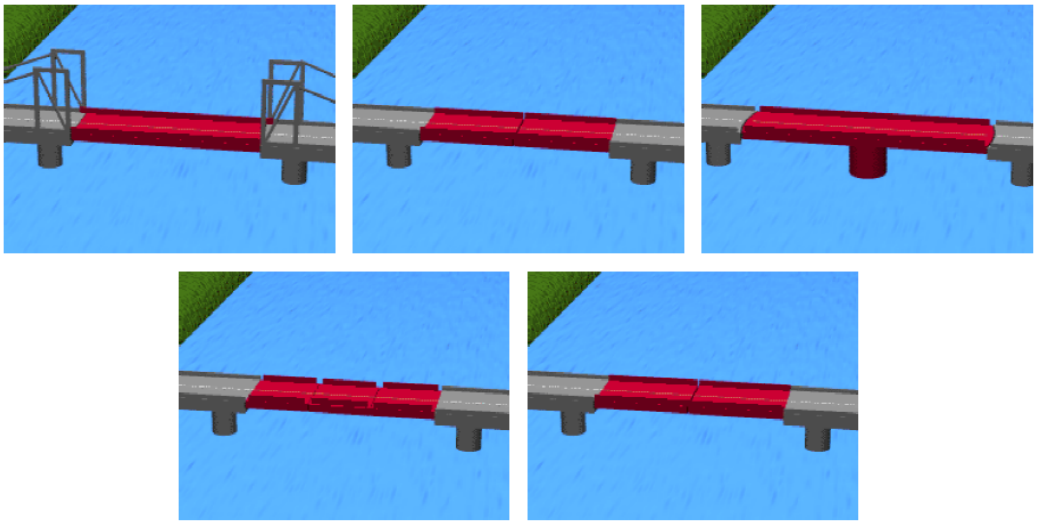

图 47. 移动桥示例（来源：ISO 6707）。

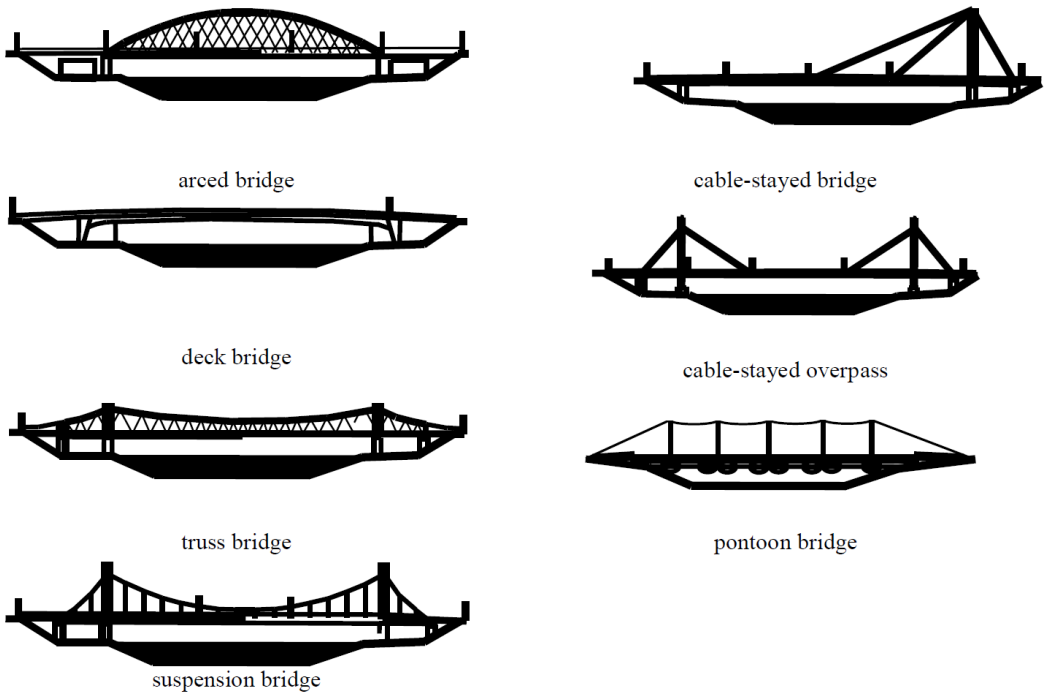

图 48. 不同类型桥梁的示例。

XML 命名空间

CityGML桥梁模块的XML命名空间由统一资源标识符（URI）标识http://www.opengis.net/citygml/bridge/2.0. 在*Bridge*模块的xmlschema定义中，这个URI还用于标识默认名称空间。

10.5.1. 桥梁和桥梁部件

BridgeType, Bridge

```
<xs:complexType name="BridgeType">
    <xs:complexContent>
        <xs:extension base="AbstractBridgeType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfBridge"
minOccurs="0" maxOccurs="unbounde</xs:sequence>
</xs:extension></xs:complexContent></xs:complexType><!--
================================================================================
=============== --><xs:element name="Bridge" type="BridgeType"
substitutionGroup="_AbstractBridge"/><!--
================================================================================
=============== --><xs:element name="_GenericApplicationPropertyOfBridge"
type="xs:anyType" abstract="true"/>
```

BridgePartType, BridgePart

```
<xs:complexType name="BridgePartType">
    <xs:complexContent>
        <xs:extension base="AbstractBridgeType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfBridgePart"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="BridgePart" type="BridgePartType" substitutionGroup=
"_AbstractBridge"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridgePart" type=
"xs:anyType" abstract="true"/>
```

如果桥的某些部分在属性值方面与其他桥不同，或者坡道等部件可以标识为它们自己的对象，则这些部分可以表示为*BridgePart*。一个桥可以由多个桥部件组成。与Bridge一样，*BridgePart*也是 *_AbstractBridge*的一个子类，因此具有相同的属性和关系。*consistOfBridgePart*表示桥（或桥部件）及其桥部件之间的聚合层次结构。通过这种方法，可以对任意深度的聚合层次结构进行建模。每个*BridgePart*只属于一个*Bridge*（或*BridgePart*）。与建筑模型类似，桥梁的聚合结构形成一棵结构树。带有零件的桥其中的一个简单示例是双桥。第10.5.6章给出了另一个例子。

AbstractBridgeType, *_AbstractBridge*

```
<xs:complexType name="AbstractBridgeType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractSiteType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="yearOfConstruction" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="yearOfDemolition" type="xs:gYear"
minOccurs="0"/>
                <xs:element name="isMovable" type="xs:boolean" default=
"false" minOccurs="0"/>
                <xs:element name="lod1Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
```

```
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="outerBridgeConstruction" type=
"BridgeConstructionElementPropertyType" minOccurs="0" maxOccurs="
unbounded"/>
                <xs:element name="outerBridgeInstallation" type=
"BridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorBridgeInstallation" type=
"IntBridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="lod3Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="interiorBridgeRoom" type=
"InteriorBridgeRoomPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="consistsOfBridgePart" type=
"BridgePartPropertyType" minOccurs="0" maxOccurs
                <xs:element name="address" type="core:AddressPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfAbstractBridge" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
```

```
</xs:complexType><!--
==========================================================================
=============== -->
<xs:element name="_AbstractBridge" type="AbstractBridgeType" abstract=
"true" substitutionGroup="core:_Site"/>
<!--
==========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAbstractBridge" type=
"xs:anyType" abstract="true
```

抽象类*AbstractBridge*是*Bridges*和*BridgeParts*的基类。它包含桥属性，纯几何表示以及桥或桥零件在不同详细程度上的几何/语义表示的属性。属性描述：

a) 桥梁或桥梁部分（*class*）的分类，不同的预期用途（*function*）和不同的实际用途（*usage*）。这些属性类型的数值可以在代码列表中指定。

b) 桥梁或桥梁部分的建造年份（*yearOfConstruction*）和拆除年份（*yearOfDemolition*）。这些属性可用于描述城市模型中桥梁发展的年表。时间点指的是真实世界的时间。

c) 桥是否可移动由Boolean属性*isMovable*指定。

10.5.2. 桥梁建造要素和桥梁装置

BridgeConstructionElementType, BridgeConstructionElement

```
<xs:complexType name="BridgeConstructionElementType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod1Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod2Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod3Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
```

```
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBridgeConstructionElement" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="BridgeConstructionElement" type=
"BridgeConstructionElementType" substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeConstructionElement"
type="xs:anyType"/>
```

BridgeFurnitureType, BridgeFurniture

```
<xs:complexType name="BridgeFurnitureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBridgeFurniture" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="BridgeFurniture" type="BridgeFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeFurniture" type=
"xs:anyType" abstract="true"/>
```

不具有*BridgePart*大小，影响或意义的桥梁图元可以建模为*BridgeConstructionElement*或*BridgeFurniture*。从结构的角度来看，必不可少的构件被建模为*BridgeConstructionElement*，例如塔架，锚具等结构构件（参见图49）。构造元素的一般分类以及预期和实际功能由属性*class*, *function* 和 *usage*表示。可能存在于LOD1到LOD4中的*BridgeConstructionElement*的几何图形为*gml:_Geometry*。或者，几何体可以作为*ImplicitGeometry*。根据隐式几何的概念，桥梁构造原型的几何模型仅在局部坐标系中存储一次，可通过其桥梁构造元素特征进行调用（参见第8.2章）。桥梁结构元素的可见表面可以使用边界表面的概念进行语义分类（参见第10.5.3章）。

尽管*BridgeConstructionElement*具有结构相关性，但*BridgeFurniture*代表桥梁的一个构件，可以在不倒塌桥梁的情况下消除该构件（如楼梯，天线，栏杆）。*BridgeFurniture*仅出现在LOD 2至4中，其几何表示为*gml:_Geometry*。同样，隐式几何的概念也可以应用于桥梁设施，其可见表面可以使用边界表面的概念进行语义分类（参见第10.5.3章）。*BridgeFurniture*包含语义属性*class*，*function*和*usage*。属性类给出了桥梁安装的分类。通过属性功能和用法，可以描述*BridgeFurniture*的名义功能和实际功能。所有属性的类型为*gml:CodeType*，其值可以在代码列表中定义。

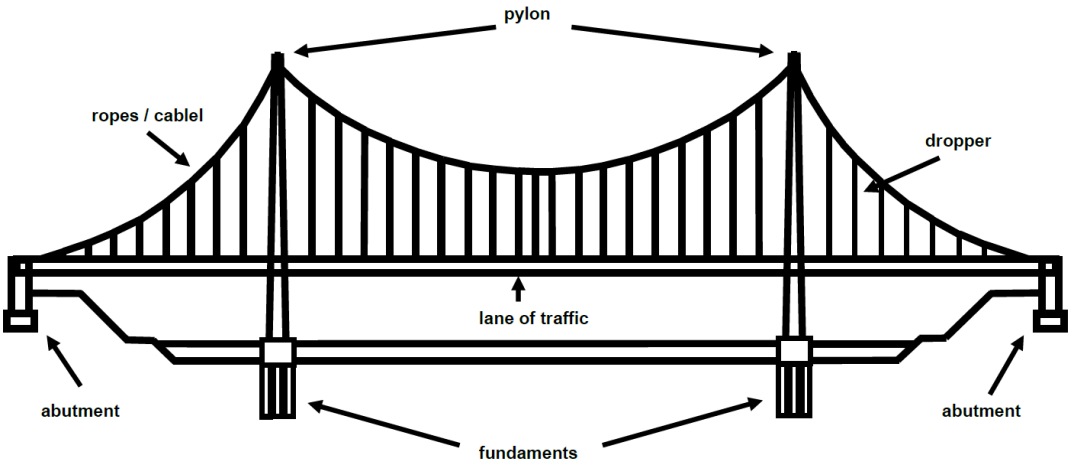

图 49. 桥梁结构悬索桥的构件。

10.5.3. 边界表面

AbstractBoundarySurfaceType, _BoundarySurface

```
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="opening" type="OpeningPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBoundarySurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
"xs:anyType" abstract=
```

桥梁的主题边界表面与建筑模块相似 *_BoundarySurface*是多个主题类的抽象基类，用于构造桥梁的外表面以及房间，桥梁构造元素和桥梁内外装置的可见表面。它是 *_CityObject*的一个子类，因此继承了所有属性，如GML3标准特征属性(*gml:name* etc.)以及CityGML特有的属性，比如外部引用。从边界表面出发，延伸出了屋顶表面，墙面，地面，外墙面，外地板表面，封闭表面，地板表面，内墙面和天花板表面等专题类。

对于Lod2到Lod4之间的每个等级， *_BoundarySurface*的几何体可以由不同的*gml:MultiSurface* 进行构建。

在LOD3和LOD4中， *_BoundarySurface*包含门和窗之类的 *_Openings* （参见第10.5.4章）。如果 *_Openings*的几何位置在拓扑上位于*gml:MultiSurface*的曲面组件内（例如 *gml:Polygon*），这些开口则表示为该曲面内的孔。孔由相应曲面几何体对象内的内环表示。根据GML3，这些点必须以相反的顺序指定（当从曲面法向量的相反方向看时，外部边界为逆时针方向，内部边界为顺时针方向）。如果此类开口由*Door*，*Window* 或*ClosureSurface*密封，则其外边界由与周围表面的内环（表示孔），其相同的点组成。*Opening*的边界表面属于相邻边界表面。例如，如果门密封了开口，则门的一侧属于*InteriorWallSurface*，另一侧属于*WallSurface*。

图 50 描绘了具有 *RoofSurfaces*，*WallSurfaces*，*OuterFloorSurfaces* 和 *OuterCeilingSurfaces* 的桥梁。 除了 *Bridges* 和 *BridgeParts* 之外，*BridgeConstructionElements*，*BridgeFurnitures* 以及 *IntBridgeFurnitures* 也都可以与 *_BoundarySurface* 相关。 *_BoundarySurfaces* 出现在 LOD2 到 LOD4 中。 在 LOD3 和 LOD4 中，该表面可能包含 *_Openings*（参见第 10.3.4 章），如门和窗。

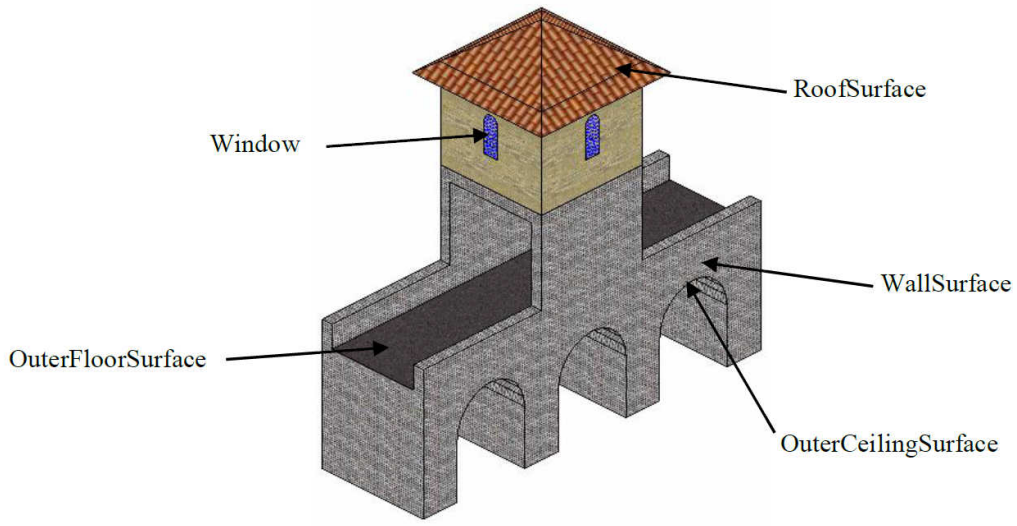

图 50. 桥的不同边界表面。

GroundSurfaceType, GroundSurface

```
<xs:complexType name="GroundSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfGroundSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true
```

桥梁或桥梁零件的接地板面由 *GroundSurface* 进行建模。定义接地板的多边形与桥梁底部轮廓线一致。但是，接地板面的表面法线指向下方。

OuterCeilingSurfaceType, OuterCeilingSurface

```
<xs:complexType name="OuterCeilingSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterCeilingSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
=============== -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true
```

属于外桥表面，且方向朝下的大部分水平表面可以建模为*OuterCeilingSurface*。

WallSurfaceType, WallSurface

```
<xs:complexType name="WallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWallSurface"
minOccurs="0" maxOccurs="unbounded</xs:sequence>
</xs:extension></xs:complexContent></xs:complexType><!--
================================================================================
============== --><xs:element name="WallSurface" type="WallSurfaceType"
substitutionGroup="_BoundarySurface"/><!--
================================================================================
============== --><xs:element name=
"_GenericApplicationPropertyOfWallSurface" type="xs:anyType" abstract=
"true"/>
```

属于外桥表面的桥梁立面，其所有部分都可以通过*WallSurface*进行建模。

OuterFloorSurfaceType, OuterFloorSurface

```
<xs:complexType name="OuterFloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfOuterFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
=============== -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
```

属于外桥表面且方向朝上的大部分水平表面，可以建模为*OuterFloorSurface*。

RoofSurfaceType, RoofSurface

```
<xs:complexType name="RoofSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfRoofSurface"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
========================================================================
================ -->
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
========================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true"/>
```

桥梁或桥梁部分的主要屋顶部分由*RoofSurface*类表示。

ClosureSurfaceType, ClosureSurface

```
<xs:complexType name="ClosureSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfClosureSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
========================================================================
=============== -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="
```

没有被门或窗填充的桥梁开口可以被虚拟表面（*ClosureSurface*）进行密封（参见第 6.4 章）。 因此，通

过虚拟封闭具有开口的桥，可以计算它们的体积。*ClosureSurfaces*也用于内部桥梁模型。 如果两个不同的房间在没有分隔门的情况下直接连接，则应使用*ClosureSurface*来分隔或连接两个房间的体积。

FloorSurfaceType, FloorSurface

```
<xs:complexType name="FloorSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref="
_GenericApplicationPropertyOfFloorSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
"xs:anyType" abstract="true"/>
```

*FloorSurface*只能在 LOD4桥梁内部模型中，用于对桥梁房间的地板进行建模。

InteriorWallSurfaceType, InteriorWallSurface

```
<xs:complexType name="InteriorWallSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfInteriorWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract="
```

*InteriorWallSurface*只能在 LOD4 桥梁内部模型中，用于对桥梁房间墙壁的可见表面进行建模。

CeilingSurfaceType, CeilingSurface

```
<xs:complexType name="CeilingSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfCeilingSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=======================================================================
============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true
```

10.5.4. 开口

AbstractOpeningType, _Opening

```
<xs:complexType name="AbstractOpeningType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfOpening"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
```

_Opening 是抽象基类，用于在语义上描述门或窗等外边界表面或内边界表面（如墙壁和屋顶）中的开口。开口仅存在于 LOD3 或 LOD4 模型中。 每个 *_Opening* 都与一个 *gml:MultiSurface* 几何体相关联。或者，其几何形体可以作为*ImplicitGeometry*对象。 根据*ImplicitGeometry*的概念，开口的原型几何，仅需要在局部坐标系中存储一次，并通过其特征进行调用（见第8.2章）。

WindowType, Window

```
<xs:complexType name="WindowType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWindow"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
```

*Window*用于建模桥梁外表皮中的窗口或相邻房间之间的舱口。*Window*和*Door*的形式上的区别在于，在正常情况下*Window*并不是专门用来运送人或车辆的。

DoorType, Door

```
<xs:complexType name="DoorType">
    <xs:complexContent>
        <xs:extension base="AbstractOpeningType">
            <xs:sequence>
                <xs:element name="address" type="core:AddressPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="_GenericApplicationPropertyOfDoor"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
=============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
```

*Door*主要用于在桥的外表皮中或相邻房间之间，对门进行建模。人们可以用门进出桥或房间。

与*ClosureSurface*不同的是，*Door*可能是关闭的，阻碍了人们的通行。一扇门可以分配多个地址（也可以不进行分配）。相应的*Address*属性类型在CityGML核心模块中定义（参见第10.1.4章）。

10.5.5. 桥梁内部

BridgeRoom，*IntBridgeFurniture*和*BridgeFurniture*表示桥梁内部构造。它们的设计类似于建筑模块中的*Room*, *IntBuildingInstallation* 和 *BuildingFurniture*。桥梁内部只能在LOD4中建模。

BridgeRoomType, BridgeRoom

```
<xs:complexType name="BridgeRoomType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="interiorFurniture" type=
"InteriorFurniturePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="bridgeRoomInstallation" type=
"IntBridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="_GenericApplicationPropertyOfBridgeRoom"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="BridgeRoom" type="BridgeRoomType" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeRoom" type=
"xs:anyType" abstract="true"/>
```

*BridgeRoom*是一个语义对象，用于对桥梁内部的空间进行建模，并且仅与一个桥梁或桥梁部件对象相关。该对象本身是闭合的（如果需要的话，可以使用*ClosureSurfaces*），并且该几何体通常使用实体（*lod4Solid*）来描

述。然而，如果不能保证边界的拓扑正确性，几何体性质也可以是多曲面（*lod4MultiSurface*）。在为*BridgeRoom*曲面指定外观时，必须将GML实体外表皮的表面法线指向外部。在这种情况下，纹理和颜色必须放置在相应曲面的背面，以便从房间内部可见。

除了几何表示之外，房间可见表面的不同部分可以由特定的*BoundarySurfaces*（*FloorSurface*, *CeilingSurface*, *InteriorWallSurface*, 和 *ClosureSurface*）进行建模；参见第10.5.3章）。

BridgeFurnitureType, BridgeFurniture

```
<xs:complexType name="BridgeFurnitureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfBridgeFurniture" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
================================================================================
============== -->
<xs:element name="BridgeFurniture" type="BridgeFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeFurniture" type=
"xs:anyType" abstract="true"/>
```

*BridgeRooms*包括*BridgeFurnitures*， *IntBridgeFurnitures*。*BridgeFurniture*是房间的可移动部分，如椅子或家具。*BridgeFurniture*对象应仅与一个房间对象相关。它的几何体可以由显式几何体或隐式几何体对象表示。根据隐式几何的概念，桥梁设施的原型几何模型仅需要在局部坐标系统中存储一次，并通过其特征进行调用（见第8.2章）。

IntBridgeFurnitureType, IntBridgeFurniture

```
<xs:complexType name="IntBridgeInstallationType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundarySurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref=
"_GenericApplicationPropertyOfIntBridgeInstallation" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType><!--
=============================================================================
============== -->
<xs:element name="IntBridgeInstallation" type="IntBridgeInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfIntBridgeInstallation"
type="xs:anyType" abstract="true"/
```

*IntBridgeFurniture*是桥中具有特定功能或语义的对象。与*BridgeFurniture*不同的是，*IntBridgeFurniture*永久附着在桥梁结构上，不能移动。例如楼梯，栏杆和加热器。*IntBridgeFurniture*的对象可以与房间（*BridgeRoom*）相关联，也可以与完整的桥梁/桥梁部件相关联（*_AbstractBridge*，参见第10.5.1章）。但是，它们应该仅与一个房间或一个桥梁/桥梁部件对象相关。*IntBridgeFurniture*具有*class*，*function*和*usage*属性。*class*只能出现一次，它表示桥梁内部组件的一般分类。通过*function*和*usage*属性，可以描述桥梁安装的名义和实际功能。对于上述三个属性，其值可以在代码列表中进行指定。对于*IntBridgeFurniture*的几何形体表示，可以使用图9所示的GML子集中的任意几何对象。或者，几何体可以作为隐式几何体对象。根据隐式几何的概念，*IntBridgeFurniture*的几何模型原型仅需要在局部坐标系中存储一次，并通过其特征进行调用（见第8.2章）。*IntBridgeFurniture*的可见表面可以使用边界表面的概念进行语义分类（参见10.5.3）。

10.5.6. 示例

横跨德国莱茵河的里斯桥有三个桥段，由桥塔隔开。图51（左）描绘了包含一个桥梁特征的Rees桥梁模型，该桥梁特征由三个*BridgePart*特征组成。桥塔在结构上是必不可少的，*BridgeConstructionElement*s来表示。在

塔架的顶部，有四个灯具，它们被建模为*BridgeFurniture*特征（参见图51的右部分）。

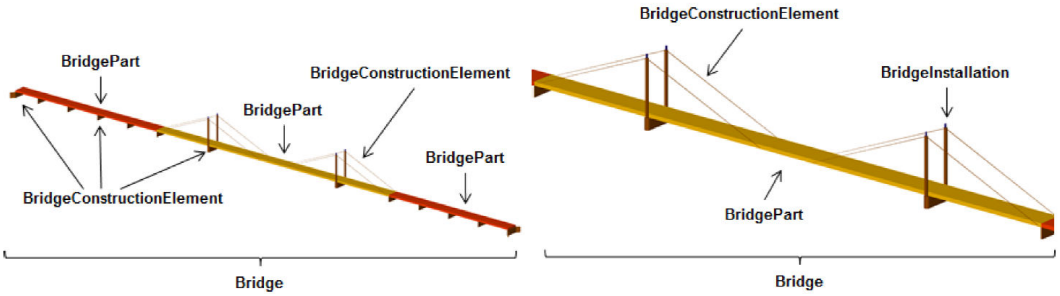

图 51. Rees桥梁，由一个桥梁特征和三个桥梁部件特征（左）组成。桥包含BridgeConstructionElement和BridgeFurniture功能（右）。

在下面的图52中，Rees桥梁的主要部分在左侧如照片所示（来源：Harald Halfpapp），而LOD2桥模型的相应部分在右侧如图所示（来源：Recklinghausen/KIT区）。

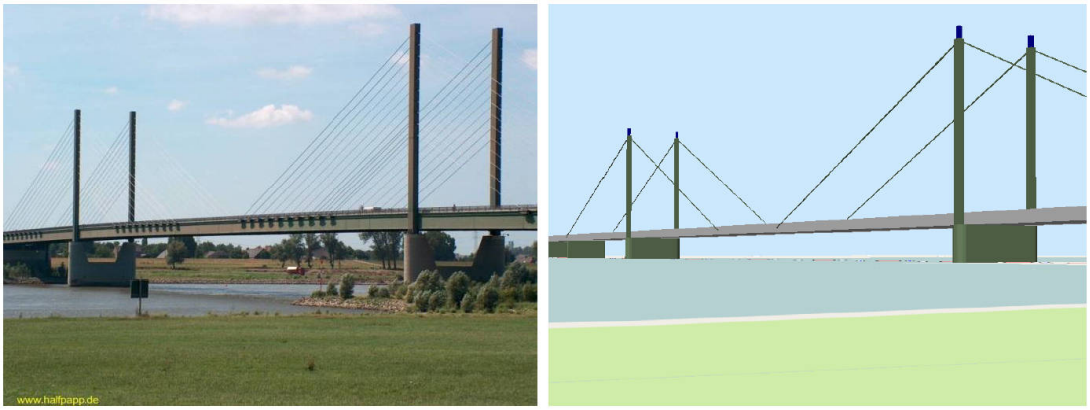

图 52. Rees桥（左图（来源：Harald Halfpap）；右图LOD2模型（来源：Recklinghausen区/KIT））。

德国卡尔斯鲁厄有两座桥横跨莱茵河。第一座是一座双轨铁路桥，建造成桁架桥（见图53）。第二座是一座四车道公路桥，作为斜拉桥建造（见图53）。

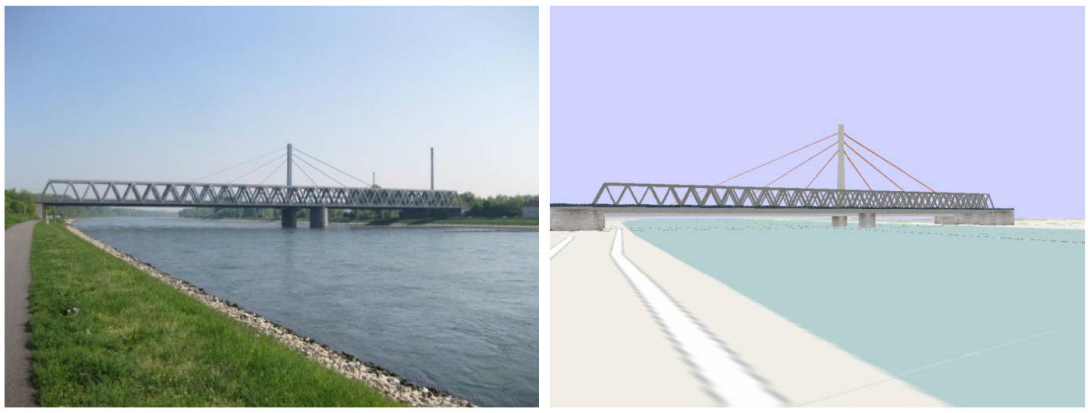

图 53. 卡尔斯鲁厄莱茵河大桥（左照片，右3D CityGML模型）（来源：KIT，卡尔斯鲁厄市提供）。

在CityGML中，这两座桥都被建模为具有*BridgeConstructionElements*和*BridgeInstallations*的单个*Bridge*对象。斜拉桥的建造要素是河流两侧和河流中间的基脚，以及缆索和桥塔。桁架桥的建造元素是基脚和桁架本身。两座桥都有几个栏杆，这些栏杆被建模为*BridgeFurniture*。

奥伯本布尔大桥如图54中所示，其位于柏林中心，横渡斯普雷河，用作具有内部房间的桥梁示例。图54的左侧描绘了真实世界的桥梁，而右侧显示了相应的CityGML模型。桥梁的外部几何结构用*gml:MultiSurface*进行建模（为*lod4MultiSurface*），并对其指定为照片级真实感纹理。此外，位于两个桥塔中的内部房间用实体几何形体表示为*BridgeRoom*对象(*lod4Solid*）。由于其高几何精度和对两个桥塔内部结构的表示，该模型被归类为LOD4。

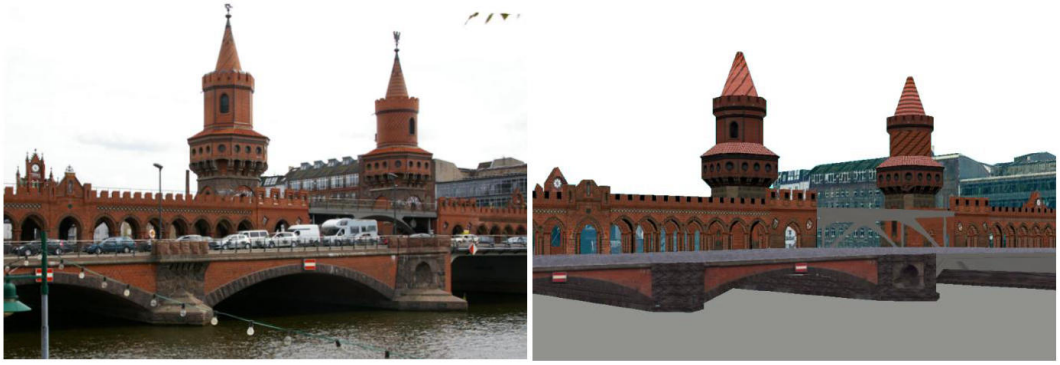

图 54. 柏林的“Oberbaumbrücke”桥在LOD4中被表示为桥梁模型（左照片，右为3D CityML模型）（来源：柏林商业，技术和妇女参议院；柏林商业定位中心；柏林理工大学；KIT）。

10.5.7. 代码列表

_AbstractBridge，*BridgeConstructionElement*，*BridgeInstallation*，*BridgeRoom*，*BridgeFurniture*和*IntBridgeFurniture*特性的*class*，*function*和*usage*属性，指定为*gml:CodeType*. 这些属性的值可以在代码列表中进行枚举。有关相应代码清单的建议见附件C.3。

10.5.8. 一致性要求

基础要求

如果一个桥只由一个部分组成，它应该用Bridge来表示。但是，如果桥梁由单独的结构段组成，则应将其建模为具有一个或多个BridgePart的Bridge。只有桥梁主体的几何和非空间特性才能在聚合Bridge元素中表示。

基于不同lod的桥梁模型构件使用限制

2. *lodXSolid*和*lodXMultiSurface*， *_AbstractBridge*的属性(*gml:SolidPropertyType*，*gml:MultiSurfacePropertyType*)可以使用几何信息对每个LOD中桥梁的外表皮进行表示（作为体量或表面模型）。对于LOD1，必须使用*lod1Solid*或*lod1MultiSurface*，但不能同时使用两者。从LOD2开始，这两个属性可以单独使用并进行建模。
3. 从LOD2开始， *_AbstractBridge*的外表皮可以使用 *_AbstractBridge*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*），对该属性在语义上分解为 *_BoundarySurface*元素。只允许将*RoofSurface*,*WallSurface*,*GroundSurface*,*OuterCeilingSurface*,*OuterFloorSurface*和*ClosureSurface*作为 *_BoundarySurface*的子类。*boundedBy*属性（注意：不要与*gml:boundedByproperty*混淆)。如果桥梁仅以LOD1表示，则不得使用。 当使用 *_BoundarySurface*元素，对桥梁外表面进行表示时，需要有一个额外的形态对表面模型进行表示（使用*lodXSolid*和*lodXMultiSurface*）。属性不应明确定义几何图形，但过程中必须使用XLink，对每层LOD中 *_BoundarySurface* 里*gml:MultiSurface* 元素中的相对应组件进行引用。
4. 从LOD2开始，可以使用*lodXMultiCurve*(属于 *_AbstractBridge*的属性)表示桥壳的曲线部分 [2..4]。如果桥梁仅以LOD1表示，则不得使用此属性。

5. 从LOD1开始， *_AbstractBridge*的*outerBridgeConstruction*属性（类型：*bridgeconstructionelementpropertypeType*）可用于对*BridgeConstructionElement*进行建模。*BridgeConstructionElement*只能用于表示桥梁的外部特征，这些特征不具有桥梁部件的重要性，但从结构角度来看是必不可少。

6. 从LOD2开始，*BridgeConstructionElement*元素的几何体可以使用*BridgeConstructionElement*的*boundedBy*属性，（类型：*BoundarySurfacePropertyType*）通过*BoundarySurface*元素进行语义分类。如果桥梁结构元素仅在LOD1中表示，则不得使用*boundedBy*属性（注意：不要与*gml:boundedByproperty*混淆)。

7. 从LOD2开始， *_AbstractBridge*的*outerBridgeFurniture*属性（类型：*BridgeFurniturePropertyType*）可用于为*BridgeFurniture*元素进行建模。*BridgeFurniture*元素仅用于表示桥梁的外部特征，这些特征不具有桥梁部件的重要性，并且从结构角度来看不是必要的。

8. 从LOD2开始，可以使用*BridgeFurniture*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*），按*BoundarySurface*元素对*BridgeFurniture*元素的几何体进行语义分类。只允许将*RoofSurface*, *WallSurface*, *GroundSurface*, *OuterCeilingSurface*, *OuterFloorSurface* 和 *ClosureSurface*作为*BoundarySurface*的子类。

9. 从LOD3开始，可以使用*BoundarySurface*的*opening*属性（类型：*OpeningPropertyType*）对*BoundarySurface*元素的开口进行建模。此属性不应用于在LOD2中进行表示的 *_BoundarySurface*元素。因此，在LOD2中表示 *_BoundarySurface*的曲面几何体必须是全连接。

   *_BoundarySurface*的*opening*属性可以包含或引用 *_Opening*元素。如果 *_Opening*元素的几何位置在拓扑上位于边界曲面的曲面组件内，则开口必须表示为该曲面的内孔。 *_Opening*元件的开口面应属于相邻边界表面。

10. 从LOD4开始， *_AbstractBridge*的*interiorBridgeRoom*属性（类型：*InteriorBridgeRoomPropertyType*）可以通过*BridgeRoom*元素对桥内部的空间进行语义建模。如果桥梁仅在LOD 1–3中表示，则不得使用此属性。*BridgeRoom*元素可以使用其*lod4Solid*或*lod4MultiSurface*属性以几何形式表示为曲面或体积模型(*gml:SolidPropertyType*，*gml:MultiSurfacePropertyType*)。

    此外，桥梁内部空间可见表面的不同部分可由主题 *_BoundarySurface*元素进行建模。其中只允许将地板表面，天花板表面，内部墙面和封闭表面作为 *_BoundarySurface*的子类。如果房间的可见表面由 *_BoundarySurface*元素表示，则使用*lod4Solid*和*lod4MultiSurface*属性的体量/表皮模型，不应明确定义几何图形。过程中必须使用XLink，对 *_BoundarySurface* 里*gml:MultiSurface*元素中的相对应组件进行引用。

11. 从LOD4开始， *_AbstractBridge*的*interiorBridgeFurniture*属性（类型：*IntBridgeFurniturePropertyType*）可用于表示桥梁内部永久性连接到桥结构的不可移动对象。如果桥梁仅在LOD 1–3中表示，则不应使用*interiorBridgeFurniture*属性。此外，仅当对象无法与*BridgeRoom*元素相关联时，才应使用*interiorBridgeFurniture*属性。在后一种情况下，应使用相应*BridgeRoom*元素的*BridgeRoomInstallation*属性（类型：*IntBridgeFurniturePropertyType*）来表示对象。

12. 从LOD4开始，可以使用*IntBridgeFurniture*的*boundedBy*属性（类型：*BoundarySurfacePropertyType*）按*BoundarySurface*元素对*IntBridgeFurniture*元素的几何体进行语义分类。其中只允许将*FloorSurface*，*CeilingSurface*，*InteriorWallSurface*和*ClosureSurface*作为*_BoundarySurface*的子类。

引用一致性

13. *_AbstractBridge*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的*_BoundarySurface*元素或使用XLink对外部 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部*_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 只有*rooftsurface*，*WallSurface*，*GroundSurface*，*OuterCeilingSurface*，*OuterFloorSurface*和*closuresurface*元素才允许由 *_AbstractBridge*的*boundedBy*属性进行封装或引用。

14. *_AbstractBridge*的*outerBridgeConstruction*（类型：*BridgeConstructionElementPropertyType*）属性可以包含在内部定义过的*BridgeConstructionElement*元素或使用XLink对外部*BridgeConstructionElement*元素进行引用。在后一种情况下*outerBridgeConstruction*属性的*xlink:href* 属性只能指向外部的*BridgeConstructionElement*元素（其中外部*BridgeConstructionElement*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

15. *_AbstractBridge*的*outerBridgeFurniture*（类型：*BridgeFurniturePropertyType*）属性可以包含在内部定义过的*BridgeFurniture*元素或使用XLink对外部*BridgeFurniture*元素进行引用。在后一种情况下*outerBridgeFurniture*属性的*xlink:href* 属性只能指向外部的*BridgeFurniture*元素（其中外部*BridgeFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

16. *_AbstractBridge*的interior*BridgeFurniture*（类型：*IntBridgeFurniturePropertyType*）属性可以包含在内部定义过的*IntBridgeFurniture*元素或使用XLink对外部*IntBridgeFurniture*元素进行引用。在后一种情况下*interiorBridgeFurniture*属性的*xlink:href* 属性只能指向外部的*IntBridgeFurniture*元素（其中外部*IntBridgeFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

17. *_AbstractBridge*的*interiorBridgeRoom*（类型：*InteriorBridgeRoomPropertyType*）属性可以包含在内部定义过的*BridgeRoom*元素或使用XLink对外部*BridgeRoom*元素进行引用。在后一种情况下*interiorBridgeRoom*属性的*xlink:href* 属性只能指向外部的*BridgeRoom*元素（其中外部*BridgeRoom*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

18. *_AbstractBridge*的*consistsOfBridgePart*（类型：*BridgePartPropertyType*）属性可以包含在内部定义过的*BridgePart*元素或使用XLink对外部的*BridgePart*元素进行引用。在后一种情况下*consistsOfBridgePart*属性的*xlink:href* 属性只能指向外部的*BridgePart*元素（其中外部*BridgePart*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

19. *_AbstractBridge*的*address*（类型：*core:AddressPropertyType*）属性可以包含在内部定义过的*core:Address*元素或使用XLink对外部的*core:Address*元素进行引用。在后一种情况下*address*属性的*xlink:href* 属性只能指向外部的*core:Address*元素（其中外部*core:Address*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

20. *_BoundarySurface*的*opening*（类型：*OpeningPropertyType*）属性可以包含在内部定义过的 *_Opening*元素或使用XLink对外部的 *_Opening*元素进行引用。在后一种情况下*opening*属性的*xlink:href* 属性只能指向外部的 *_Opening*元素（其中外部 *_Opening*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

21. *Door*的*address*（类型：*core:AddressPropertyType*）属性可以包含在内部定义过的*core:Address*元素或使用XLink对外部的*core:Address*元素进行引用。在后一种情况下*address*属性的*xlink:href* 属性只能指向外部的*core:Address*元素（其中外部*core:Address*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

22. *BridgeConstru-tionElement*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部的 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

23. *BridgeFurniture*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部的 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面元素只允许被*BridgeFurniture* 的*boundedBy*属性进行封装或引用。

24. *IntBridgeFurniture*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部的 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面元素只允许被*IntBridgeFurniture* 的*boundedBy*属性进行封装或引用。

25. *BridgeRoom*的*boundedBy*（类型：*BoundarySurfacePropertyType*）属性可以包含在内部定义过的 *_BoundarySurface*元素或使用XLink对外部的 *_BoundarySurface*元素进行引用。在后一种情况下*boundedBy*属性的*xlink:href* 属性只能指向外部的 *_BoundarySurface*元素（其中外部 *_BoundarySurface*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。 屋顶表面，墙面，地表面，外墙面，外地板表面和封闭表面元素只允许被*BridgeRoom* 的*boundedBy*属性进行封装或引用。

26. *BridgeRoom*的*interiorFurniture*（类型：InteriorFurniturePropertyType）属性可以包含在内部定义过的*BridgeFurniture*元素或使用XLink对外部的*BridgeFurniture*元素进行引用。在后一种情况下*interiorFurniture*属性的*xlink:href* 属性只能指向外部的*BridgeFurniture*元素（其中外部*BridgeFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

27. *BridgeRoom*的*bridgeRoomInstallation*（类型：*IntBridgeFurniturePropertyType*）属性可以包含在内部定义过的*IntBridgeFurniture*元素或使用XLink对外部的*IntBridgeFurniture*元素进行引用。在后一种情况下*bridgeRoomInstallation*属性的*xlink:href* 属性只能指向外部的*IntBridgeFurniture*元素（其中外部*IntBridgeFurniture*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

28. *BridgeConstructionElement*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

29. *BridgeFurniture*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

30. *_Opening*的*lodXImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lodXImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

31. *IntBridgeFurniture*的*lod4ImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType* ）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

32. *BridgeFurniture*的*lod4ImplicitRepresentation*（类型：*core:ImplicitRepresentationPropertyType*）属性可以包含在内部定义过的*core:ImplicitGeometry*元素或使用XLink对外部*core:ImplicitGeometry*元素进行引用。在后一种情况下*lod4ImplicitRepresentation*属性的*xlink:href* 属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。该元素及引用，只能选择都给出，或都不给出。

10.6. 水体

水在城市化进程中一直扮演着重要的角色，城市最好建在河流旁和易于靠岸的地方。显然，水对人类的营养和卫生是必不可少的。水体提供了最经济的交通方式，同时也是阻碍直接进入其他地点的隔离带。在水道上建立桥梁，是第一次努力建设的尝试，并产生了今天的高科技桥梁。许多城市的景观都是以水为主，这直接关系到三维城市模型。此外，作为娱乐的场所和发生潜在危害（如洪水）的地方，水体对城市生活非常重要。

水体与建筑物，道路和地形的永久性不同的独特特征，将在该专题模型中得到考虑。水体是动态表面。潮汐以一定的规律出现，但不定期事件在自然力量中占据主导地位，例如洪水事件。此时能看见的水面的高度及其覆盖区域会发生变化，因此必须以一种不同于地形或建筑物等邻接对象的建模方式对其语义和几何形状进行建模。

第一种水体建模方法满足了三维城市模型的要求。它不继承任何水文或其他动态方面的信息。在这方面，它并不是完整的。然而，这里给出的语义和几何描述将允许动态的和概念上不同的描述在之后得到进一步的发展。CityGML的水体模型包含在专题扩展模块*WaterBody*中（参见第7章）。

水体模型代表河流，运河，湖泊和盆地的专题和三维几何形状方面的信息。在LOD 2-4中，水体以不同的专题表面为边界线。其中包括必须提供的*WaterSurface*，定义为水和空气之间的边界；可选的*WaterGroundSurface* ，定义为水和地下之间的边界（例如DTM或三维盆地对象的地面）；以及*WaterClosureSurface*，可包含零个或多个，定义为不同水体之间或水与某个建模区域的边界之间的虚拟边界（见图 55）。可以利用*WaterSurface*作为动态元素以表示潮滩的临时变化情况。

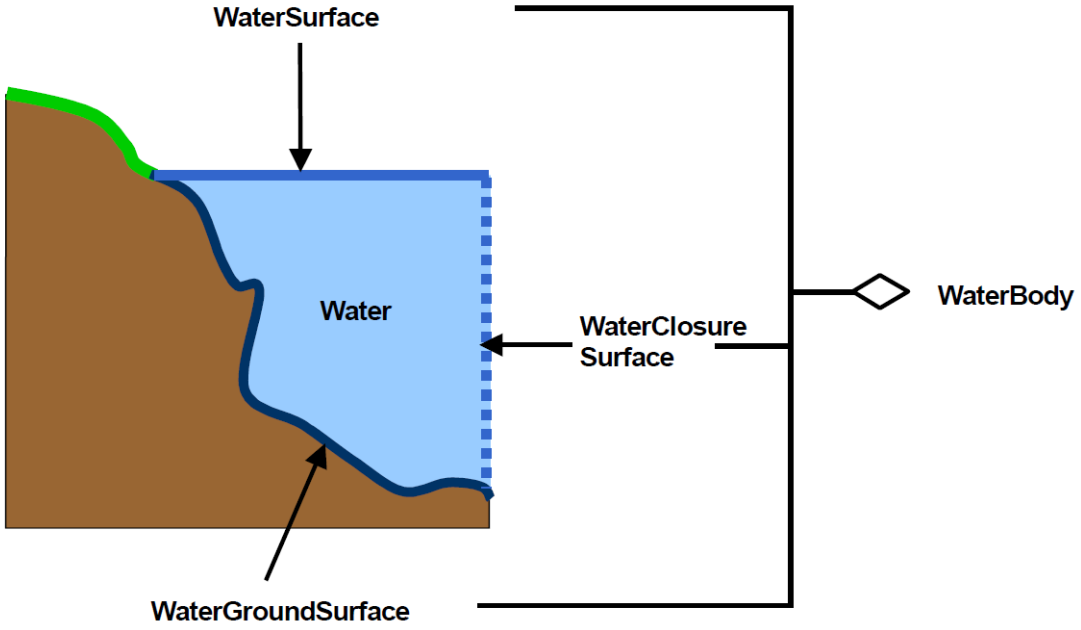

图 55. CityGML定义的水体的图示（图片来源：波恩大学大地测量和地理信息研究所）

水体模型的UML图如图 56所示，XML模式定义见下文和附录A.13。每个*WaterBody*对象具有*class*，*function*和*usage*等属性，其取值可以在代码列表中进行枚举（参见第10.6.3章和附录C.9）。*class*属性定义了对象的分类，例如湖泊，河流或喷泉，并且只能出现一次。*function*属性包含对象的目的，例如国家水道或公共泳池，而*usage*属性定义了其实际用途，例如水体是否通航。后两个属性均可多次出现。

WaterBody是WaterObject的子类，是根类 CityObject的衍生对象。将来_WaterObject类可能会在水对象的更多子类中得到细分。WaterBody的几何表达会随着不同的细节级别而具有差异。由于WaterBody是 CityObject的子类，因此它也是继承了gml:name属性的要素。WaterBody可通过 _WaterBoundarySurface类在语义上进行细分。 _WaterBoundarySurface是水体外表皮的一部分，具有部分特殊功能，如WaterSurface，WaterGroundSurface或WaterClosureSurface。与任何 _CityObject一样，WaterBody，WaterSurface，WaterGroundSurface和WaterClosureSurface都可以分配外部引用（参见第6.7章），并且可以使用CityGML的泛型模块（参见第10.12章）的泛型属性对其进行扩充。

*WaterSurface*的可选属性*waterLevel*可用于描述水位，此时必须给定相应的三维表面几何形状。当水体受潮汐影响时，这一点尤其重要。其允许的取值可以在相应的代码列表中进行定义。

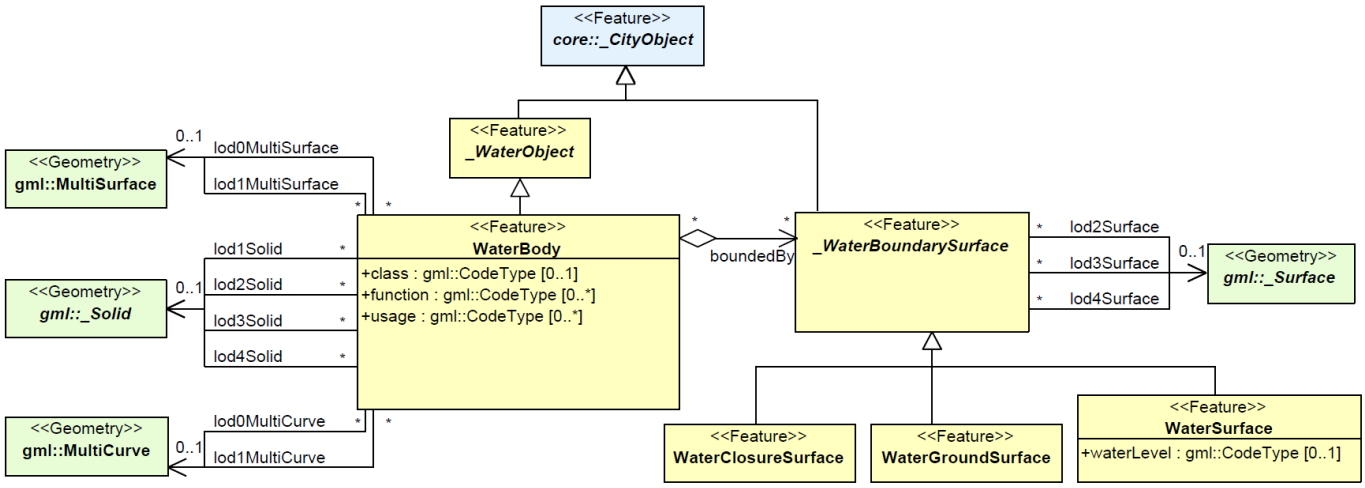

图 56. CityGML中水体模型的UML图。前缀用于指示与模型元素相关的XML命名空间。不带前缀的元素名称将在CityGML的WaterBody模块中定义。

LOD0和LOD1都只进行较低程度的的描述和较高程度的概念化。此处河流通过*MultiCurve*几何体进行建模，而小溪则被省略。具有明显范围的大海，海洋和湖泊则表示为））*MultiSurface*（图 56）。可以为每个*WaterBody*分配不同类型的几何组合。线性水体可表示为三维曲线的网络。每条曲线由直线段组成，其中线的方向表示流动方向（水流从曲线的第一个点，例如*gml:LineString*，流到最后一个点）。湖泊或海洋等面对象由水面的三维表面几何图形表示。

从LOD1开始，水体也可以以充满水的*Solids*的形式进行建模。如果水体由LOD2或更高细节级别的*gml:Solid*表示，则对应专题的*WaterClosureSurface*，*WaterGroundSurface*和*WaterSurface*对象的表面几何形状必须与*gml:Solid*的外壳重合。如果对于每个LOD X相应的*lodXSolid*（其中X介于2和4之间）没有对几何体进行冗余的表达，而是引用*WaterClosureSurface*，*WaterGroundSurface*和*WaterSurface*的*lodXSurface*元素（其中X介于2和4之间）所对应的几何元素（可运用GML3的语言特性）。

LOD2到LOD4需要更高级别的细节，因此任何*WaterBody*都可以由专题表面或由周围专题表面构成的固定实体勾勒出轮廓。

WaterSurface，*WaterClosureSurface*和*WaterGroundSurface*类的每个对象都必须至少具有一个关联的几何体。这意味着CityGML实例文档中的每个*WaterSurface*，*WaterClosureSurface*和*WaterGroundSurface*要素都必须至少包含以下属性之一：*lod2Surface*，*lod3Surface*，*lod4Surface*。

水体模型还隐含了*TerrainIntersectionCurves*(TIC) 的概念。例如，为了指定DTM与*WaterBody*的三维几何图形的精确交集，或将*WaterBody*或*WaterSurface*调整到周围的DTM上（参见第6.5章）。此时*WaterSurface*多边形的环则暗示了水体与地形或盆地的交集。

XML命名空间

CityGML的*WaterBody*模块的XML命名空间由统一资源标识符（URI）标识 http://www.opengis.net/citygml/waterbody/2.0 。在*WaterBody*模块的XML模式定义中，此URI还用于标识默认命名空间。

10.6.1. 水体

AbstractWaterObjectType, _WaterObject

```
<xs:complexType name="AbstractWaterObjectType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfWaterObject"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_WaterObject" type="AbstractWaterObjectType" abstract=
"true" substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterObject" type=
"xs:anyType" abstract="true"/>
```

WaterBodyType, WaterBody

```
<xs:complexType name="WaterBodyType">
    <xs:complexContent>
        <xs:extension base="AbstractWaterObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod0MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod0MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1MultiCurve" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod2Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod3Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="lod4Solid" type="gml:SolidPropertyType"
minOccurs="0"/>
                <xs:element name="boundedBy" type=
"BoundedByWaterSurfacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="_GenericApplicationPropertyOfWaterBody"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
==========================================================================
============== -->
<xs:element name="WaterBody" type="WaterBodyType" substitutionGroup=
"_WaterObject"/>
<!--
==========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterBody" type="
xs:anyType" abstract="true"/>
```

10.6.2. 边界表面

根据不同的功能和特性，为了构建固定的或复合的表面几何形状，定义了三种水边界类（图 55）。

1. “从空气到水”的边界类。*WaterSurface*对模型来说是强制性的，需要通常使用摄影测量分析或地图探索的方式进行注册。其几何表达可能因潮滩或水位变化而有所不同。这可以通过包含具有不同*waterLevels*（*gml:CodeType*）的不同静态水面来反映，例如最高洪水事件，平均海平面，或最低水位等级。上述内容提供了描述较为重要的水面的机会，通常这些水位对于类似于潮汐带等现象的表达而言很重要。
2. “从水到地面”的边界类。*WaterGroundSurface*可以通过声纳探测或其他深度测量方式获取。“水到建筑物”的边界也是地表面的组成部分。地表可能与水下地形模型相同，但也描述了其他水下物体的轮廓。该概念的用处源于水闸，防水石，防洪堤，或潮汐发电站等水防建筑的存在。使用*WaterGroundSurface*作为与人造建筑物的边界层，在城市环境中经常出现。此类对象可能完全包裹着建模得到的水体，例如喷泉和游泳池。*WaterSurface*对象与*WaterGroundSurface*对象共同把*WaterBody*围合成一个体量模型。
3. “从水到水”的边界类。*WaterClosureSurface*是一个可选要素。当水体的*WaterSurfaces*和*WaterGroundSurfaces*的并集未能定义为封闭的体量时，该要素将得以使用。然后*WaterClosureSurface*将用作实现水体量的围合，并将水体量与仅具有表面的水体量分开。该功能可以在河流的横截面和地表在其流经部分仅有部分信息可获取的情况下使用。

_*WaterBoundarySurfaces*应仅作为对应的*WaterBody*对象的一部分，不应成为CityGML模型中的独立对象。

AbstractWaterBoundarySurfaceType, _WaterBoundarySurface

```
<xs:complexType name="AbstractWaterBoundarySurfaceType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="lod2Surface" type=
"gml:SurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3Surface" type=
"gml:SurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4Surface" type=
"gml:SurfacePropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfWaterBoundarySurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="_WaterBoundarySurface" type=
"AbstractWaterBoundarySurfaceType" abstract="true" substitutionGroup=
"core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterBoundarySurface" type
="xs:anyType" abstract="true"/>
```

WaterSurfaceType, WaterSurface

```
<xs:complexType name="WaterSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractWaterBoundarySurfaceType">
            <xs:sequence>
                <xs:element name="waterLevel" type="gml:CodeType"
minOccurs="0"/>
                <xs:element ref="
_GenericApplicationPropertyOfWaterSurface" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="WaterSurface" type="WaterSurfaceType" substitutionGroup
="_WaterBoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWaterSurface" type=
"xs:anyType" abstract="true"/>
```

WaterGroundSurfaceType, WaterGroundSurface

```
<xs:complexType name="WaterGroundSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractWaterBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfWaterGroundSurface" minOccurs="0" maxOccurs
="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="WaterGroundSurface" type="WaterGroundSurfaceType"
substitutionGroup="_WaterBoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterGroundSurface" type=
"xs:anyType" abstract="true"/>
```

WaterClosureSurfaceType, WaterClosureSurface

```
<xs:complexType name="WaterClosureSurfaceType">
    <xs:complexContent>
        <xs:extension base="AbstractWaterBoundarySurfaceType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfWaterClosureSurface" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="WaterClosureSurface" type="WaterClosureSurfaceType"
substitutionGroup="_WaterBoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterClosureSurface" type=
"xs:anyType" abstract="true"/>
```

10.6.3. 代码列表

*WaterBody*的*class*，*function*，*usage*属性，以及*WaterSurface*要素的*waterLevel*属性指定为*gml:CodeType*。这些属性的值可以在代码列表中进行枚举。相应的代码可在附录C.9中查找。

10.6.4. 一致性要求

基础要求

1. 对于LOD0和LOD1，可以使用*gml:MultiCurve*几何元素将*WaterBody*几何形体建模为线性网络。在这种情况下，每个*gml:MultiCurve*应由直线段组成。其中线的方向表达出流向。流向是从线段的第一个点到最后一个点。

2. 从LOD2开始，*WaterBody*的外表皮可以使用*WaterBody*的*boundedBy*属性（类型：*boundedByWaterSurfacePropertyType*）在语义上分解为多个 *_WaterBoundarySurface*元素。如果水体仅在较低级别的LOD中得到表达，则不应使用*boundedBy*属性。

   如果外表皮由 *_WaterBoundarySurface*元素表示，则不应使用*WaterBody*的*lodXSolid*，X∈[2..4]属性对水体量进行额外的几何表达，但是必须借助GML 3.1.1的XLink概念，对每层LOD中*_WaterBoundarySurface*对象里相应的*gml:MultiSurface*元素中进行引用。

3. 每个 *_WaterBoundarySurface*元素必须至少有一个与其关联的表面几何形体。该几何形体由*_WaterBoundarySurce*的*lodXSurface*，X∈[2..4]属性给出。

4. *_WaterBoundarySurface*元素只能作为相应 *WaterBody*元素的一部分。在CityGML模型中，该元素不能作为独立的城市对象。

引用一致性

5. *WaterBody*元素的*boundedBy*属性（类型：*boundedByWaterSurfacePropertyType*）可以在内部包含*_WaterBoundarySurface*元素，或借助GML 3.1.1的XLink概念，对外部的 *_WaterBoundarySurface*元素进行XLink引用。在后一种情况下*boundedBy*属性的*XLink:href*属性只能指向外部的*_WaterBoundarySurface*元素（其中外部 *_WaterBoundarySurface*元素位于另一份文档或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.7. 交通对象

CityGML的交通模型是一个多功能，多尺度的模型，侧重于专题和功能，以及几何或拓扑方面的信息。交通要素在LOD0中被表达为线性网络。从LOD1开始，所有交通要素都由三维表面进行几何描述。交通要素的面建模将使得几何路线规划算法的应用成为可能。这可用于确定交通路线上所需的限制和操作。该信息还可用于现实世界中移动机器人的轨迹规划，或在三维可视化和训练模拟器中自动放置替身（虚拟人）或车辆的模型。CityGML的交通模型由专题扩展模块*Transportation*提供（参见第7章）。

其中最主要的类为*TransportationComplex*。它可以表达道路，轨道，铁路，或广场等。图57说明了四个不同的专题类。

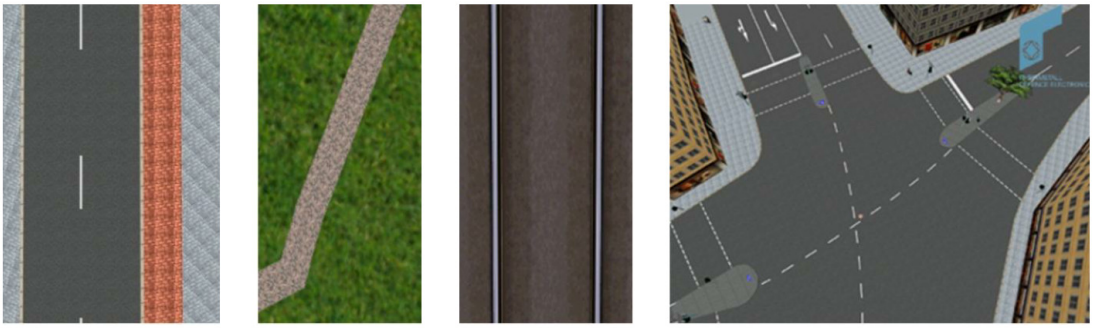

图 57. TransportationComplex类的表达(1)（图片来源：德国莱茵金属国防电子公司）

*TransportationComplex*由*TrafficArea*和*AuxiliaryTrafficArea*两部分组成。图 58描绘了虚拟三维城市模型中LOD2的*TransportationComplex*设置的示例。*Road*由几个*TrafficArea*组成，分别用于人行道，车行道，停车场，以及这些区域位于凸起花坛之下的*AuxiliaryTrafficArea*。

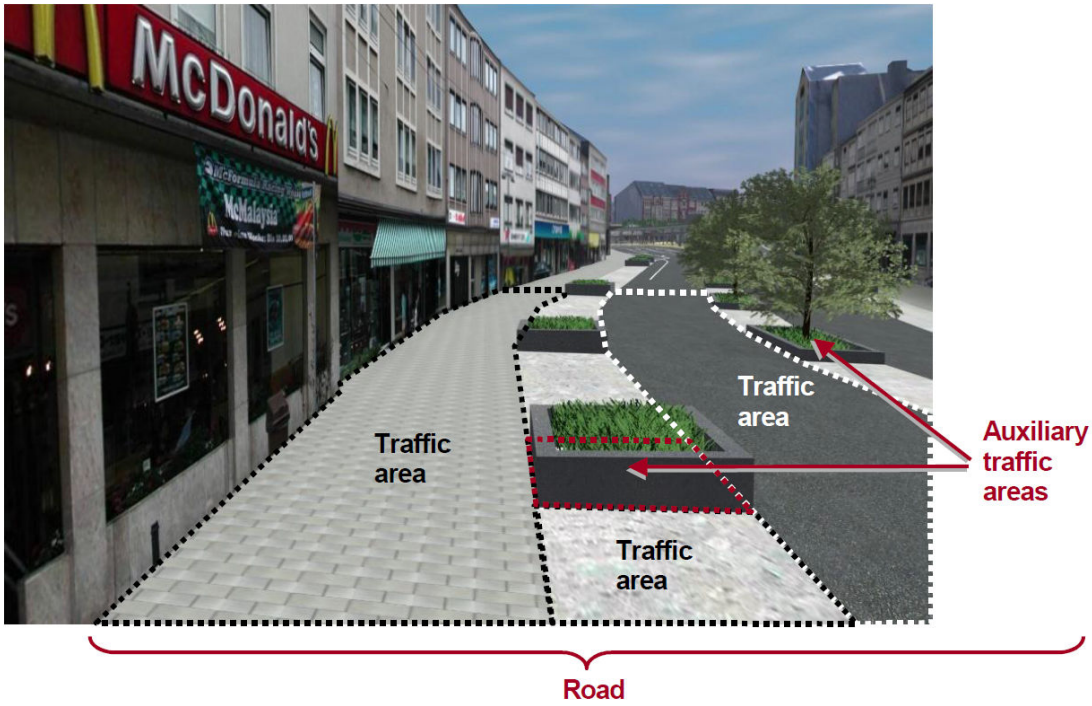

图 58. CityGML中LOD2层级的TransportationComplex类的表达示例。其中的道路是TrafficArea类和AuxiliaryTrafficArea类的集合体（图片来源：索林根市波恩大学大地测量和地理信息研究所）

图 59描述了交通模型的UML图。其中XML模式定义见附录A.10。

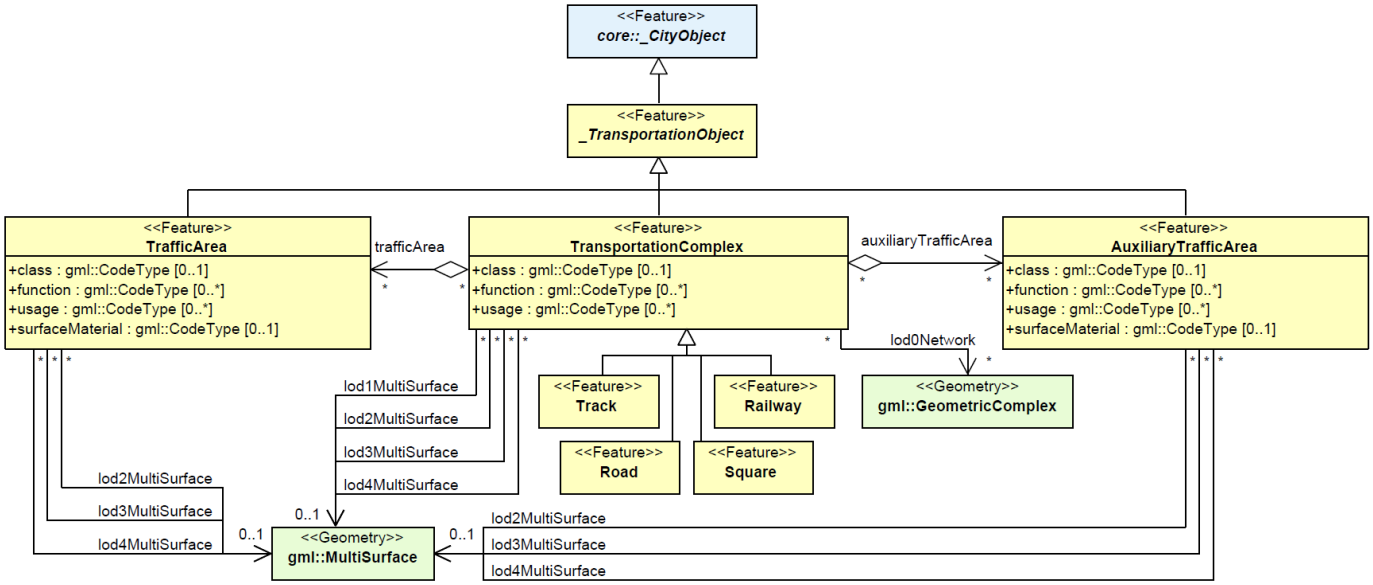

图 59. CityGML中交通模型的UML图。前缀用于指示与模型元素相关的XML命名空间。不带前缀的元素名称将在CityGML的Transportation模块中定义。

道路本身被表达为*TransportationComplex*，并可进一步细分为*TrafficArea*和*AuxiliaryTrafficArea*。*TrafficArea*是在交通方面很重要的元素，如车行道，步行区，自行车道。*AuxiliaryTrafficArea*描述了道路的其余元素，如路缘石，中间车道和绿地等。

可以通过*Track*，*Road*，*Railway*和*Square*等子对象来对*TransportationComplex*对象进行专题方面的进一步区分。每个*TransportationComplex*都有*class*，*function*和*usage*属性，其取值可以在代码列表中进行枚举（第10.7.4章和附录C.8）。*class*属性描述了对象的分类；*function*描述了对象的用途，例如国家高速公路，乡道，或机场；而如果实际用途与功能不同，则可以使用*usage*属性。*function*和*usage*属性都可以多次出现。

此外，*TrafficArea*和*AuxiliaryTrafficArea*都具有*class*，*function*，*usage*和*surfaceMaterial*属性。*class*属性描述了对象的分类。对于*TrafficArea*，*function*描述了对象是否属于车行道，步行区，还是自行车道；*usage*属性则指示哪些交通模式（例如行人，汽车，电车，旱冰鞋）可以使用它。*surfaceMaterial*属性指定路面类型，且可用于*AuxiliaryTrafficArea*（例如沥青，混凝土，砾石，土壤，铁路，草地）。*AuxiliaryTrafficArea*的*function*属性定义了其功能是否为路缘石，中间车道或绿色区域。其数值可以在代码列表中指定。

每个交通区域的形状由表面几何形状进行定义。可以通过使用预定义目录的属性来定义其他元数据。这将影响每个交通区域的*class*，*function*，*usage*及其*surfaceMaterial*。属性目录可能会因客户或国家的不同而有所不同。下表显示了各种*TrafficArea*的示例：

示例	乡道	高速公路入口
TransportationComplex - function	道路	道路
TrafficArea - usage	汽车，货车，公交车，出租车，摩托车	汽车，货车，公交车，出租车，摩托车
TrafficArea - function	行车道	高速公路入口
TrafficArea - surfaceMaterial	沥青	混凝土

示例	机场跑道	机场停机坪
TransportationComplex - function	道路	停机坪
TrafficArea - usage	飞机	飞机，汽车，货车，公交车，行人
TrafficArea - function	机场-跑道	机场-停机坪
TrafficArea - surfaceMaterial	混凝土	混凝土

*TransportationComplex*是*_TransportationObject*和根类 *_CityObject*的子类。*TransportationComplex*的几何表达因细节级别的不同会有变化。由于*TransportationComplex*是 *_CityObject*的子类，因此它继承了属性*gml:name*。街道名称也存储在*Road*要素的*gml:name*属性中。

在最粗糙的LOD0中，交通混合体通过线对象建模形成线性网络。在该抽象层次上，可实现路径查找或其他类似分析的算法。它还可以用于生成交通网络的示意图和可视化。由于交通网络这样的抽象定义不包含针对交通对象的明确描述，因此生成图形可视化界面可能是可视化应用的任务，例如，可使用已定义好每个交通对象的具体样式（宽度，颜色和纹理）的库。

从LOD1开始，*TransportationComplex*将提供明确的表面几何图形，以反映对象的实际形状，而不仅仅是它的中心线。LOD2到LOD4级别中，它将进一步按专题被细分为用于主要交通的*TrafficArea*：例如汽车，火车，公共交通，飞机，自行车或行人，以及用于对交通目的而言没那么重要的*AuxiliaryTrafficArea*：例如道路标记，绿地或花盆。每个LOD的*TransportationComplex*的不同表达方式如图60所示。

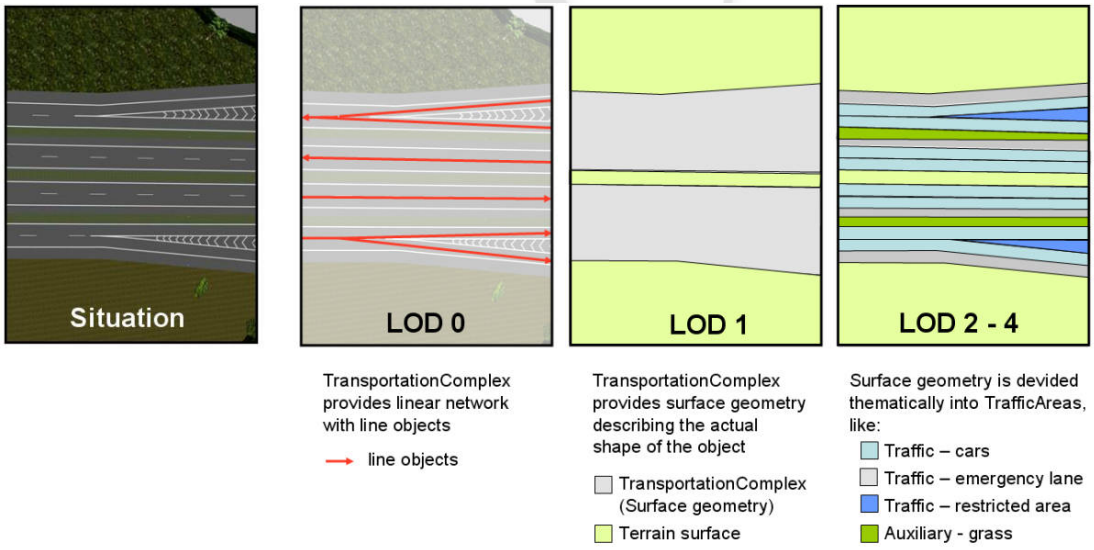

图 60. TransportationComplex类的实际情况（示例展示的是高速公路的局部）（图片来源：德国莱茵金属国防电子公司）

在LOD0中，广场等面状交通对象的建模方式应与GDF中的建模方式相同。GDF是大多数汽车导航系统所使用的交通网络ISO标准。在GDF中，广场通常使用围绕该区域的环进行表达，并且该环与邻近道路相连。CityGML并不涵盖交通网络模型所包含的功能性特征（例如速度限制），因为它旨在补充而不是取代GDF等现有标准。但是，如果特定的功能必须与CityGML交通对象发生关联，此时则可以使用CityGML的*泛型*模块（参见第10.12章）提供*泛型属性*。此外，可以通过使用*ExternalReferences*（参见第6.11章）从其他信息系统添加更多感兴趣的对象。例如，为汽车导航提供额外信息的GDF数据集可用于交通流的模拟和可视化。可以使用字典的概念（参见第6.6章）对对象的属性值进行扩充或替换。这些目录可能会随着所处国家或使用用户的不同而有所不同（特别是国家专属的道路标志和信号）。

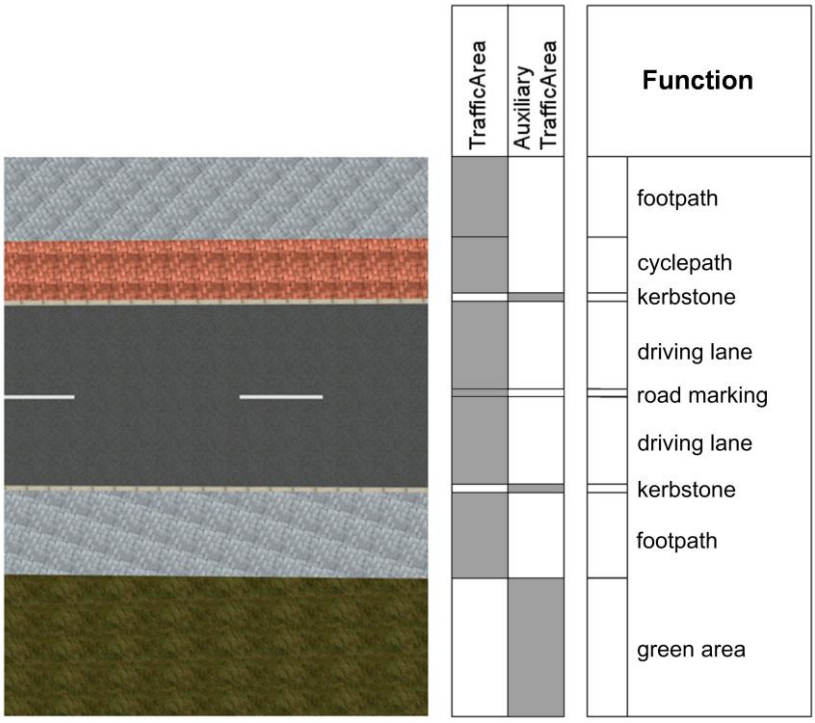

图 61. LOD2-4层级的TransportationComplex类，对具有复杂横剖面设计的道路的表达（示例展示的是城市道路）（图片来源：德国莱茵金属国防电子公司）

以下示例展示了一个复杂的城市交叉口。图62中左图是一个训练模拟器的编辑器界面的截屏。它允许定义由交通对象，外部引用，建筑物和植被对象所组成的道路网络。图62中右图展示了交叉口的三维形象，其中包括所有被引用的静态和动态模型。

图 62. 复杂的城市交叉口：带有表面描述和外部引用的线性交通网络（图片来源：德国莱茵金属国防电子公司）

XML命名空间

CityGML*Transportation*模块的XML命名空间由统一资源标识符（URI）标识 http://www.opengis.net/citygml/transportation/2.0 。在*Transportation*模块的XML模式定义中，该URI也用于标识默认命名空间。

10.7.1. 交通混合体

AbstractTransportationObjectType, _TransportationObject

```
<xs:complexType name="AbstractTransportationObjectType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfTransportationObject" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="_TransportationObject" type=
"AbstractTransportationObjectType" abstract="true" substitutionGroup=
"core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTransportationObject" type
="xs:anyType" abstract="true"/>
```

*_TransportationObject*表示交通对象的抽象基类。CityGML交通模型未来的扩展应按照该类的子类的形式进行建模。

TransportationComplexType, TransportationComplex

```
<xs:complexType name="TransportationComplexType">
    <xs:complexContent>
        <xs:extension base="AbstractTransportationObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="trafficArea" type=
"TrafficAreaPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="auxiliaryTrafficArea" type=
"AuxiliaryTrafficAreaPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="lod0Network" type=
"gml:GeometricComplexPropertyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfTransportationComplex" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="TransportationComplex" type="TransportationComplexType"
substitutionGroup="_TransportationObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTransportationComplex"
type="xs:anyType" abstract="true"/>
```

该类型和元素描述了道路或铁路等交通混合体可从不同的专题部分（即交通区域，例如人行道和辅助交通区域）组合而成。作为_ _*CityObject*的子类，*TransportationComplex*继承了其所有的属性和关系，特别是id，名称，外部引用，和泛型关系。此外，它还表达为不同专题类型的交通混合体的基类。

10.7.2. 交通混合体的子类

TrackType, Track

```
<xs:complexType name="TrackType">
    <xs:complexContent>
        <xs:extension base="TransportationComplexType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfTrack"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Track" type="TrackType" substitutionGroup=
"TransportationComplex"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTrack" type="xs:anyType"
abstract="true"/>
```

*Track*是主要供行人使用的小路。它是*TransportationComplex*的子类，因此继承了它的所有属性和关系。

RoadType, Road

```
<xs:complexType name="RoadType">
    <xs:complexContent>
        <xs:extension base="TransportationComplexType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfRoad"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Road" type="RoadType" substitutionGroup=
"TransportationComplex"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRoad" type="xs:anyType"
abstract="true"/>
```

*Road*旨在用于表达汽车等车辆使用为主的交通要素，例如街道，高速公路和乡村道路等。它是*TransportationComplex*的子类，因此继承了它的所有属性和关系。

RailwayType, Railway

```
<xs:complexType name="RailwayType">
    <xs:complexContent>
        <xs:extension base="TransportationComplexType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfRailway"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Railway" type="RailwayType" substitutionGroup=
"TransportationComplex"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRailway" type="xs:anyType"
abstract="true"/>
```

*Railway*表示有轨电车或火车等轨道车辆所使用的路线。它是*TransportationComplex*的子类，因此继承了它的所有属性和关系。

SquareType, Square

```
<xs:complexType name="SquareType">
    <xs:complexContent>
        <xs:extension base="TransportationComplexType">
            <xs:sequence>
                <xs:element ref="_GenericApplicationPropertyOfSquare"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Square" type="SquareType" substitutionGroup=
"TransportationComplex"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfSquare" type="xs:anyType"
abstract="true"/>
```

*Square*是城市中常见的开放区域（例如广场，集市广场）。它是*TransportationComplex*的子类，因此继承了它的所有属性和关系。

10.7.3. 交通混合体的细分

TrafficAreaType, TrafficArea

```
<xs:complexType name="TrafficAreaType">
    <xs:complexContent>
        <xs:extension base="AbstractTransportationObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="surfaceMaterial" type="gml:CodeType"
minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfTrafficArea"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="TrafficArea" type="TrafficAreaType" substitutionGroup=
"_TransportationObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTrafficArea" type=
"xs:anyType" abstract="true"/>
```

AuxiliaryTrafficAreaType,AuxiliaryTrafficArea

```
<xs:complexType name="AuxiliaryTrafficAreaType">
    <xs:complexContent>
        <xs:extension base="AbstractTransportationObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="surfaceMaterial" type="gml:CodeType"
minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfAuxiliaryTrafficArea" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="AuxiliaryTrafficArea" type="AuxiliaryTrafficAreaType"
substitutionGroup="_TransportationObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfAuxiliaryTrafficArea" type
="xs:anyType" abstract="true"/>
```

10.7.4. 代码列表

TransportationComplex，*TrafficArea*和*AuxiliaryTrafficArea*要素的*class*，*function*和*usage*属性，以及*TrafficArea*和*AuxiliaryTrafficArea*要素的*surfaceMaterial*属性都被指定为*gml:CodeType*。这些属性的值都可以在代码列表中进行枚举。相应的代码列表都可在附录C.8中查找。

10.7.5. 一致性要求

基础要求

1. 对于LOD0，*TransportationComplex*的几何形状应使用GML线对象进行建模，以便表达出交通混合体的中心线。这些线对象应构成一个线性网络。因此，*TransportationComplex*元素的*lod0Network*属性（类型：*gml:GeometricComplexPropertyType*）可能只包含或只引用合适的曲线几何元素。

2. 从LOD2开始，可以使用*TransportationComplex*元素的*trafficArea*属性（类型：*TrafficAreaPropertyType*）以及*AuxiliaryTrafficArea*属性（类型：*AuxiliaryTrafficAreaPropertyType*）。如果交通混合体仅以较低的LOD级别表达，则不应使用这些属性。

引用一致性

3. *TransportationComplex*元素的*trafficArea*属性（类型：*TrafficAreaPropertyType*）可以在内部包含*TrafficArea*元素，或借助GML 3.1.1的XLink概念，对外部的*TrafficArea*元素进行XLink引用。在后一种情况下，*trafficArea*属性的*XLink:href*属性只能指向外部的*TrafficArea*元素（其中外部的*TrafficArea*元素位于另一份文档或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

4. *TransportationComplex*元素的*AuxiliaryTrafficArea*属性（类型：*TrafficAreaPropertyType*）可以在内部包含*AuxiliaryTrafficArea*元素，或借助GML 3.1.1的XLink概念，对外部的*AuxiliaryTrafficArea*元素进行XLink引用。在后一种情况下，*AuxiliaryTrafficArea*属性的*XLink:href*属性只能指向外部的*AuxiliaryTrafficArea*元素（其中外部*AuxiliaryTrafficArea*元素位于另一份文档或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.8. 植被对象

植被要素是三维城市模型的重要组成部分，因为它们可辅助对周围环境的识别。对植被对象进行分析和可视化后，便可对其分布，结构和多样性进行研究，并可对动植物栖息地进行分析，推导出动植物群所受到的影响。植被模型可用作针对诸如森林火灾模拟，城市曝气模拟或微气候模拟的基础。该模型还可用于检测森林破坏，检测障碍物（例如和航空交通相关的障碍），或环境保护领域的分析任务等。CityGML的植被模型由专题扩展模块*Vegetation*定义（参见第7章）。

CityGML的植被模型对树木等独立植被对象，和诸如森林或其他植物群落等生物群落的植被区（图 63）进行了区分。单个植被对象由*SolitaryVegetationObject*类建模，而对于填充了特定植被的区域，则使用*PlantCover*类。*PlantCover*要素的几何形态可以是*MultiSurface*或*MultiSolid*。这取决于该植被的垂直范围。例如，对于森林，使用*MultiSolid*表达可能更合适。植被模型的UML图如图 64所示，XML模式定义见下文和附录A.12。

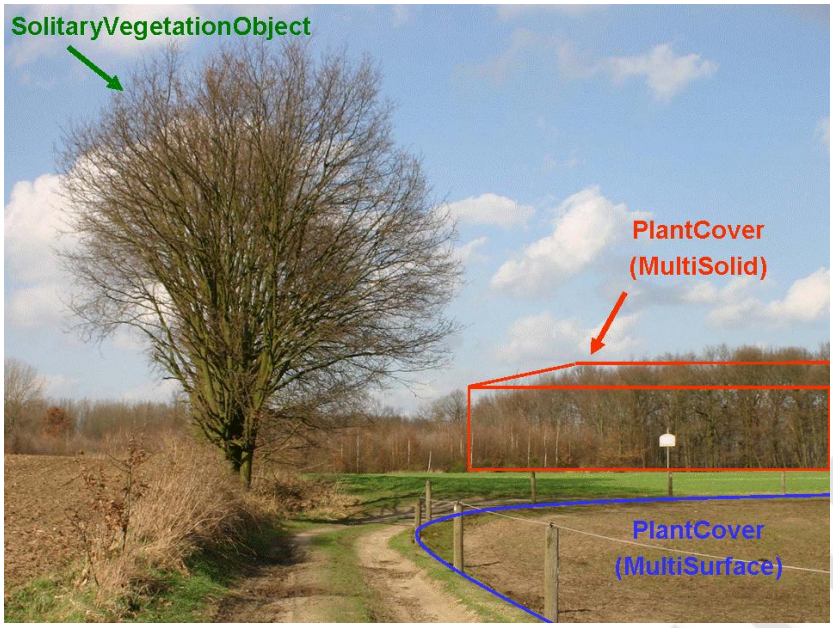

图 63. SolitaryVegetationObject类和PlantCover类的植被对象示例（图片来源：Recklinghausen）

*SolitaryVegetationObject*可能具有*class*，*function*，*usage*，*species*，*height*，*trunkDiameter*和*crownDiameter*属性。*class*属性包含了对象或植物习性的分类，例如树，灌木，草，并且只能出现一次（见第10.8.4章和附录C.7）。*species*定义了物种的名称，例如*Abies alba*，并且最多只能出现一次（参见第10.8.4章和附录C.7）。可选属性*function*和*usage*表示对象的预期真实用途，例如植物园，并且可以多次出现。*class*，*species*，*function*和*usage*的属性值可以在代码列表中定义。*height*属性包含对象的相对高度。*crownDiameter*和*trunkDiameter*属性分别表示植物冠和树干直径。树干直径常用于城市的地籍法规（例如树木管理规则）。

*PlantCover*要素可能具有*class*，*function*，*usage*和*averageHeight*属性。*PlantCover*的植物群落由*class*属性表示。该属性的值可以在代码列表中指定（参见第10.8.4章和附录C.7）。其取值不仅可以描述单一的植物类型或物种，还可以表示植物群落中植物类型常见组合。该信息可用于生成逼真的三维可视化效果，其中*PlantCover*区域会自动填充相应的（可能随机）混组的三维植物对象。*function*，*usage*属性指代着对象的预期真实目的，例如国家森林，并且可以多次出现。*averageHeight*属性表示植被的相对平均高度。

由于*SolitaryVegetationObject*和*PlantCover*都是从*_ _CityObject*类继承而来的，它们继承了城市对象的的所有属性，尤其是名称（*gml:name*）和指向外部信息系统对应对象的*ExternalReference*，其中可能包含来自公共环境机构的植物信息（见第6.7章）。

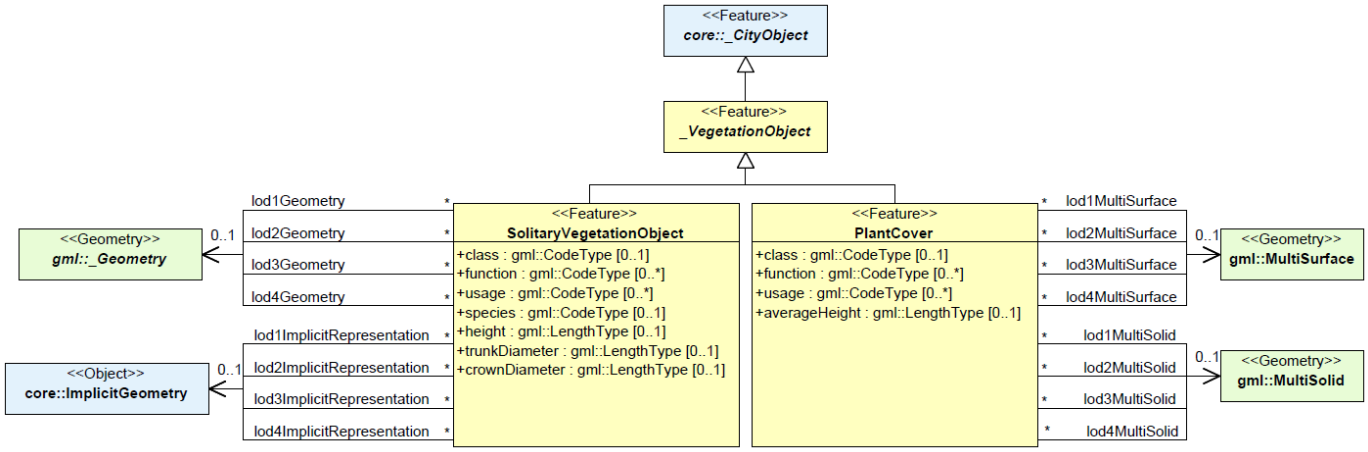

图 64. CityGML中植被对象的UML图。前缀用于指示与模型元素相关的XML命名空间。不带前缀的元素名称将在CityGML的Vegetation模块中定义。

*SolitaryVegetationObject*的几何图形可以在LOD 1-4中由具有绝对坐标值的GML几何图形明确定义，或者通过*__ImplicitGeometry__*原型进行定义（参见第8.2章）。独立植被对象可能是在隐式几何体适用的情形下最典型的例子之一，因为对于大多数的植被对象类型而言，例如同一物种的树木，在大多数情况下都可以认为其形状是相同的。

此外，与季节变化相关的植物外观可以使用*__ImplicitGeometry__*进行表达。出于可视化的目的，只有定义了形状和外观的对象数据库的内容需要被替换（如图65所示）。

图 65. 植被对象在不同季节的可视化（1）（图片来源：Recklinghausen）

*SolitaryVegetationObject*或*PlantCover*类对象在每个LOD中可能都具有不同的几何体。其中*SolitaryVegetationObject*与表达任意GML几何体的*gml:_Geometry*类相关联（通过关系*lodXGeometry*，X∈[1..4]）；*PlantCover*只使用*gml:MultiSolid*或*gml:MultiSurface*进行建模。*PlantCover*的其中一个示例是通过*gml:MultiSolid*建模的实体森林模型，参见图 66。

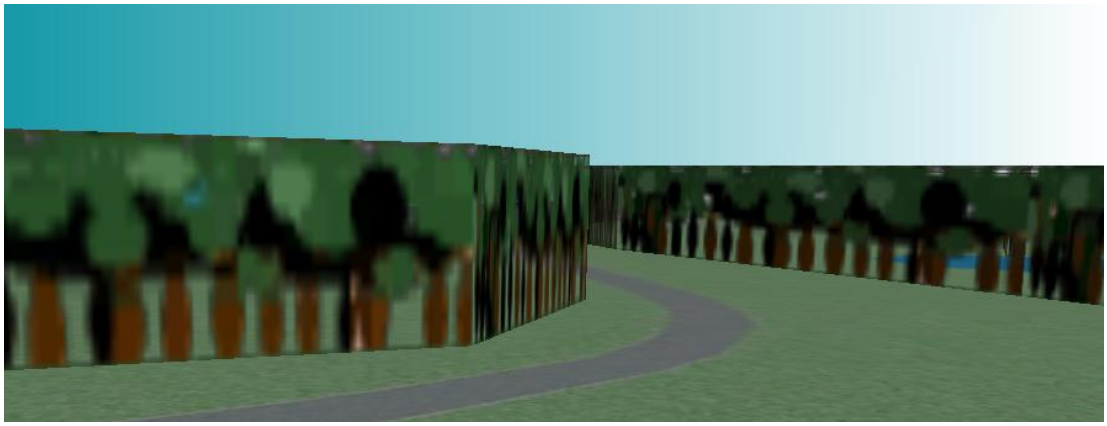

图 66. 森林实体的可视化/建模示例（图片来源：Recklinghausen区）

XML命名空间

CityGML的*植被*模块的XML命名空间由统一资源标识符 (URI)标识 http://www.opengis.net/citygml/vegetation/2.0 。在*植被*模块的XML模式定义中，此URI还用于标识默认命名空间。

10.8.1. 植被对象

AbstractVegetationObjectType, _VegetationObject

```
<xs:complexType name="AbstractVegetationObjectType" abstract="true">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element ref=
"_GenericApplicationPropertyOfVegetationObject" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_VegetationObject" type="AbstractVegetationObjectType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfVegetationObject" type=
"xs:anyType" abstract="true"/>
```

10.8.2. 孤立植被对象

SolitaryVegetationObjectType, SolitaryVegetationObject

```
<xs:complexType name="SolitaryVegetationObjectType">
```

```
    <xs:complexContent>
        <xs:extension base="AbstractVegetationObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="species" type="gml:CodeType" minOccurs=
"0"/>
                <xs:element name="height" type="gml:LengthType" minOccurs
="0"/>
                <xs:element name="trunkDiameter" type="gml:LengthType"
minOccurs="0"/>
                <xs:element name="crownDiameter" type="gml:LengthType"
minOccurs="0"/>
                <xs:element name="lod1Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod2Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod3Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfSolitaryVegetationObject" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
=========================================================================
=============== -->
<xs:element name="SolitaryVegetationObject" type=
"SolitaryVegetationObjectType" substitutionGroup="_VegetationObject"/>
<!--
=========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfSolitaryVegetationObject"
```

```
type="xs:anyType" abstract="true"/>
```

10.8.3. 植被覆盖对象

PlantCoverType, PlantCover

```
<xs:complexType name="PlantCoverType">
    <xs:complexContent>
        <xs:extension base="AbstractVegetationObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="averageHeight" type="gml:LengthType"
minOccurs="0"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1MultiSolid" type=
"gml:MultiSolidPropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiSolid" type=
"gml:MultiSolidPropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSolid" type=
"gml:MultiSolidPropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSolid" type=
"gml:MultiSolidPropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfPlantCover"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
==========================================================================
============== -->
<xs:element name="PlantCover" type="PlantCoverType" substitutionGroup=
"_VegetationObject"/>
<!--
==========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfPlantCover" type=
"xs:anyType" abstract="true"/>
```

10.8.4. 代码列表

*PlantCover*和*SolitaryVegetationObject*要素的*class*，*function*和*usage*属性，以及*SolitaryVegetationObject*要素的*species*属性被指定为*gml:CodeType*。这些属性的值可以在代码列表中进行枚举。相应的代码列表可在附录C.7中查找。

10.8.5. CityGML数据集示例

以下是CityGML数据集的两个片段，包含一棵独立树（*SolitaryVegetationObject*）和一个植物群落（*PlantCover*）。独立树具有以下属性：*class*=1070（落叶树），*species*=1040（水青冈/山毛榉），*height*=8米，*trunckDiameter*=0.7米，*crownDiameter*=8.0米。植物群落具有以下属性：*class*=1180（菊苣苔属），*averageHeight*=0.5m。*class*和*species*的属性取自SIG 3D提出的代码列表，见附录C.7。

```
<SolitaryVegetationObject>
    <class codeSpace=
"http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetation
Object_class.xml">1070</class>
    <species codeSpace=
"http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetation
Object_species.xml">1040</species>
    <height uom="#m">8</height>
    <trunkDiameter uom="#m">0.7</trunkDiameter>
    <crownDiameter uom="#m">8</crownDiameter>
    <lod1ImplicitRepresentation>
        <core:ImplicitGeometry>
            <core:mimeType>1010</core:mimeType>
            <core:libraryObject>urn:sig3d:tree.wrl</core:libraryObject>
            <core:referencePoint>
                <gml:Point srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
                    <gml:pos srsDimension="3">5733690.578 2571129.123
60.0</gml:pos>
                </gml:Point>
            </core:referencePoint>
        </core:ImplicitGeometry>
    </lod1ImplicitRepresentation>
</SolitaryVegetationObject>
```

```
<PlantCover>
    <class codeSpace=
"http://www.sig3d.org/codelists/standard/vegetation/2.0/PlantCover_class.x
ml">1180</class>
    <averageHeight uom="#m">0.5</averageHeight>
    <lod1MultiSurface>
        <gml:MultiSurface>
            <gml:surfaceMember>
                <gml:Polygon srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
                    <gml:exterior>
                        <gml:LinearRing>
                            <gml:pos srsDimension="3">5733806.146
2571329.227 60.0</gml:pos>
                            <gml:pos srsDimension="3">5733754.782
2571387.011 60.0</gml:pos>
                            <gml:pos srsDimension="3">5733674.527
2571374.170 60.0</gml:pos>
                            <gml:pos srsDimension="3">5733670.246
2571274.653 60.0</gml:pos>
                            <gml:pos srsDimension="3">5733764.413
2571243.621 60.0</gml:pos>
                            <gml:pos srsDimension="3">5733806.146
2571329.227 60.0</gml:pos>
                        </gml:LinearRing>
                    </gml:exterior>
                </gml:Polygon>
            </gml:surfaceMember>
        </gml:MultiSurface>
        …
    </lod1MultiSurface>
</PlantCover>
```

10.8.6. 一致性要求

引用一致性

*SolitaryVegetationObject*元素的*lodXImplicitRepresentation*，X∈[1..4]（类型：*core:ImplicitRepresentationPropertyType*）属性可以在内部包含一个*core:ImplicitGeometry*元素，或借助GML 3.1.1的XLink概念，对外部的*core:ImplicitGeometry*元素进行XLink引用。在后一种情况下,*lodXImplicitRepresentation*，X∈[1..4]属性的*XLink:href*属性只能指向外部的*core:ImplicitGeometry*元素（其中外部的*core:ImplicitGeometry*元素位于另一份文档，或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.9. 城市家具

城市家具是不可移动的物体，如路灯，红绿灯，交通标志，花盆，广告栏，长凳，定界桩或公共汽车站（如

图67和68所示）。城市家具对象可以在交通区，住宅区，广场或建筑区找到。城市家具对象的建模用于可视化，例如城市交通，但也用于分析当地的结构条件。使用这些精细建模的城市家具对象，可以促进对城市模型中特殊位置的识别，城市模型本身也变得更加生动。CityGML的城市家具模型由主题扩展模块*CityFurniture*定义（参见第7章）。

例如，城市家具对象可以对城市交通状况的模拟产生重要影响。导航系统因此得以实现，例如，视觉障碍的人可使用交通灯作为导航目标。同样，城市家具对象对于重型车辆运输的规划非常重要。它们必须知道障碍物的确切位置以及更准确的相关信息。

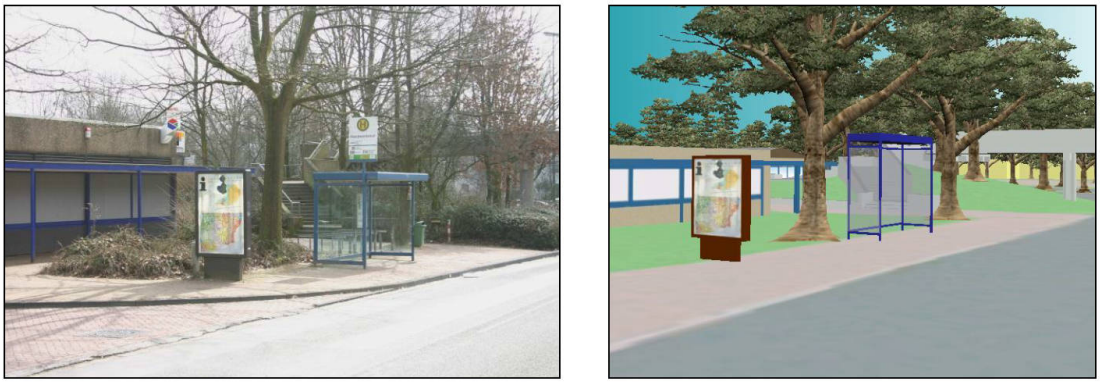

图 67. 广告牌和庇护所以CityFurniture对象的形式建模（来源：德国Barkenberg的三维城市模型）

图 68. 真实场景中的灯笼和定界桩（来源：德国Barkenberg的三维城市模型）

城市家具模型的UML图如图 69所示，XML模式定义见下文和附录A.5。

*CityFurniture*可以具有*class*，*function*和*usage*属性。其值可在相应的代码列表中规定（第10.9.2章和附录C.4）。*class*属性用于对对象进行分类，如红绿灯，交通标志，定界桩或垃圾桶，并且只需要定义一次。*function*属性描述城市家具对象所属的主题领域（例如交通，交通管制，建筑），并且可以多次出现。*usage*属性表示对象的真正用途。

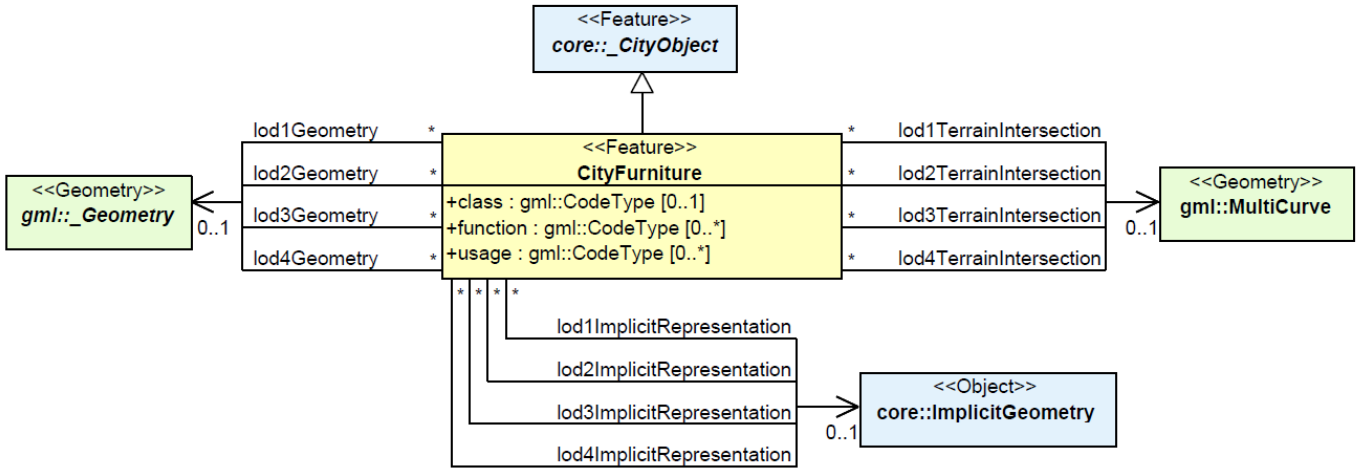

图 69. CityGML中城市家具对象的UML图。前缀用于指示与模型元素相关的XML命名空间。不带前缀的元素名称在CityGML的CityFurniture模块中定义。

由于*CityFurniture*是*_ CityObject的子类，它是一个要素，它因此继承了gml:name属性。同样地，与_CityObject*一样，*CityFurniture*对象可以指定*外部引用*（参见第6.7章），并且可以使用CityGML*泛型*模块（参见第10.12章），通过*泛型属性*进行扩充。对于*外部引用*，城市家具对象可以链接到外部主题数据库。因此，不能在CityGML中建模的对象，其语义信息可以在三维城市模型中传输并用于进一步处理，例如来自电力线路或管道系统，交通标志地籍或用于灾害管理的水资源的信息。

城市家具对象可以用特定的几何图形在城市模型中进行表示，但在大多数情况下，同一类型对象具有相同的几何图形。LOD1-4中*CityFurniture*对象的几何图形可以由显式几何图形（*lodXGeometry*，其中*X*介于1和4之间）或*__ImplicitGeometry对象*（lodXImplicitRepresentation，*其中X介于1和4之间）表示。根据ImplicitGeometry__*的概念，城市家具原型的几何信息仅需要在局部坐标系中存储一次，并被其他城市家具要素调用（见第8.2章）。例如，可以从城市地图或公私外部信息系统中获取城市家具对象的空间信息。

为了指定DTM与城市家具对象的三维几何体的精确交集，后者可以为每个LOD设置*TerrainIntersectionCurve*（TIC）（参见第6.5章）。这将使得DTM和城市家具对象之间的过渡变得顺畅。

XML命名空间

CityGML *CityFurniture*模块的XML命名空间由统一资源标识符（URI）进行标识 http://www.opengis.net/citygml/cityfurniture/2.0 。 在*CityFurniture*模块的XML模式定义中，该URI还被用于识别默认的命名空间。

10.9.1. 城市家具对象

CityFurnitureType, CityFurniture

```
<xs:complexType name="CityFurnitureType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
```

```
maxOccurs="unbounded"/>
                <xs:element name="lod1Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod2Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod3Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfCityFurniture" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="CityFurniture" type="CityFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityFurniture" type=
"xs:anyType" abstract="true"/>
```

10.9.2. 代码列表

*CityFurniture*要素的*class*，*function*和*usage*属性被指定为*gml:CodeType*。这些属性的值将在代码列表中进行列举。相应代码列表可在附录C.4中查找。

10.9.3. CityGML示例数据集

下面的CityGML示例数据集是LOD3中定界桩模型的片段，包含属性*class*=1000和*function*=1520（定界桩）。其编码属性值取自SIG 3D提出的代码列表（参见附录C.4）。对象ID为*stake0815*的定界桩，有一个*外部引用*指向德国ALKIS数据库中的地籍对象 (http://www.adv-online.de)，可由其URI标识进行识别：*urn:adv:oid:DEHE123400007001*。

示例数据集显示了上表面（*gml:id "cover"*）和左表面（*gml:id "surfLeft"*）的几何图形。图 70对其进行了描述。上表面指定了一个固定的材质（白色），左表面通过指示相关材质的坐标的方式使用纹理图像*stack.gif*进行纹理设置。两种表面的外观都使用CityGML的*外观*模块进行建模（参见第9章）。

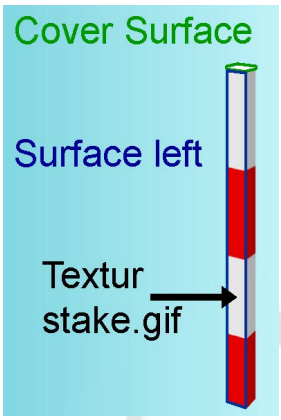

图 70. 一个简单的城市家具对象的示例（来源：Recklinghausen区）

```
<!-- delimitation stake in LOD3 -->
<CityFurniture gml:id="stake0815">
    <core:externalReference>
        <core:informationSystem>http://www.adv-
online.de</core:informationSystem>
        <!-- Reference to ALKIS -->
        <core:externalObject>
            <core:uri>urn:adv:oid:DEHE123400007001</core:uri>
            <!-- ALKIS Object ID -->
        </core:externalObject>
    </core:externalReference>
    <app:appearance>
        <app:Appearance>
            <app:surfaceDataMember>
                <app:X3DMaterial>
                    <app:ambientIntensity>0.4</app:ambientIntensity>
                    <app:diffuseColor>1.0 1.0 1.0</app:diffuseColor>
                    <app:target>#cover</app:target>
                </app:X3DMaterial>
            </app:surfaceDataMember>
            <app:surfaceDataMember>
                <app:ParameterizedTexture>
                    <app:imageURI>stake.gif</app:imageURI>
                    <app:textureType>typical</app:textureType>
```

```
                    <app:target uri="#surfLeft">
                        <app:TexCoordList>
                            <app:textureCoordinates ring="#surfLeft_ring1
"> 0.000 0.000 1.000 0.000 1.000 1.000 0.000 1.000 0.000 0.000
</app:textureCoordinates>
                        </app:TexCoordList>
                    </app:target>
                </app:ParameterizedTexture>
            </app:surfaceDataMember>
        </app:Appearance>
    </app:appearance>
    <class codeSpace=
"http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_c
lass.xml">1000</class>
    <!-- 1000 = traffic -->
    <function codeSpace=
"http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_f
unction.xml">1520</function>
    <!-- 1520 = boundary post -->
    <lod3Geometry>
        <!--Stake 0.06x0.06x1.2-->
        <gml:Solid srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
            <gml:exterior>
                <gml:CompositeSurface>
                    <gml:surfaceMember>
                        <gml:Polygon gml:id="cover">
                            <gml:exterior>
                                <!-- Cover-Surface -->
                                <gml:LinearRing>
                                    <gml:pos>5733500.060 2572400.060
61.200</gml:pos>
                                    <gml:pos>5733500.060 2572400.000
61.200</gml:pos>
                                    <gml:pos>5733500.000 2572400.000
61.200</gml:pos>
                                    <gml:pos>5733500.000 2572400.060
61.200</gml:pos>
                                    <gml:pos>5733500.060 2572400.060
61.200</gml:pos>
                                </gml:LinearRing>
                            </gml:exterior>
                        </gml:Polygon>
                    </gml:surfaceMember>
                    <gml:surfaceMember>
                        <gml:Polygon gml:id="surfLeft">
                            <gml:exterior>
                                <!-- Surface left -->
                                <gml:LinearRing gml:id="surfLeft_ring1">
```

```
                                    <gml:pos>5733500.060 2572400.000
60.000</gml:pos>
                                    <gml:pos>5733500.000 2572400.000
60.000</gml:pos>
                                    <gml:pos>5733500.000 2572400.000
61.200</gml:pos>
                                    <gml:pos>5733500.060 2572400.000
61.200</gml:pos>
                                    <gml:pos>5733500.060 2572400.000
60.000</gml:pos>
                                </gml:LinearRing>
                            </gml:exterior>
                        </gml:Polygon>
                    </gml:surfaceMember>
                    ...
                </gml:CompositeSurface>
            </gml:exterior>
        </gml:Solid>
    </lod3Geometry>
</CityFurniture>
```

10.9.4. 一致性要求

引用一致性

- *CityFurniture*元素的*lodXImplicitRepresentation*（X ∈ [1..4]）（类型：*core:ImplicitRepresentationPropertyType*）属性可能包含内部定义的*core:ImplicitGeometry*元素，或使用GML 3.1.1的XLink概念对外部的*core:ImplicitGeometry*元素进行XLink引用。在后一种情况下，*lodXImplicitRepresentation*（X ∈ [1..4]）属性的*XLink:href*属性只能指向外部的*core:ImplicitGeometry*元素（其中外部*core:ImplicitGeometry*元素位于另一个文档或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.10. 土地利用

*LandUse*对象可以用来描述用于特定土地利用类型的地球表面区域，也可以用来描述具有土地覆盖的地球表面区域。这些土地覆盖的区域既包括植被覆盖的区域，也包括未被植被覆盖的区域，如沙地，岩石，泥滩，森林，草地等（即实体外表）。土地利用和土地覆盖是两个不同的概念；前者描述地球表面的人类活动，后者描述地球表面的物理和生物意义上的覆盖。然而，这两者是相互联系，且在实践中往往是相互融合的。CityGML中的*LandUse*对象代表了这两个概念：它们可以用来表示地块，与空间规划相关的对象，与休闲相关的对象，在三维层面描述某区域物理特征的对象（例如湿地）。图 71显示了土地利用对象的UML图。XML模式定义见第10.10.1章和附录A.8。CityGML的土地利用模型由*LandUse*专题扩展模块提供（参见第7章）。

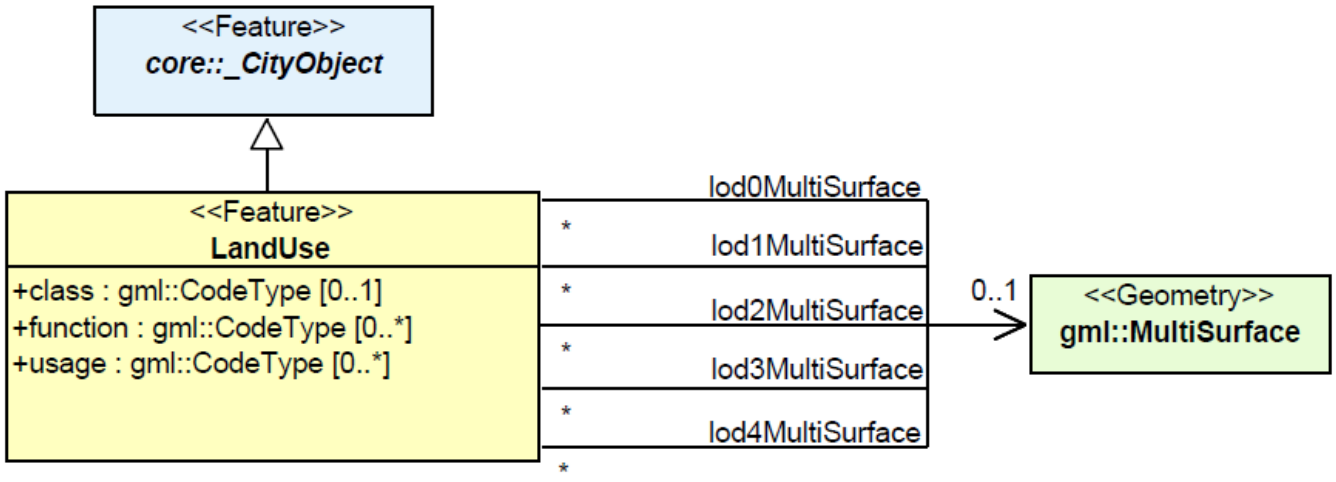

图 71. CityGML土地利用对象的UML图。前缀用于表示与模型元素相关的XML命名空间。没有前缀的元素名字在CityGML的LandUse模块中定义。

每个*LandUse*对象都具有*class*，*function*和*usage*属性。*class*属性用于表示土地使用对象的分类，如聚落区，工业区，农田等，且只能出现一次。代码列表中指定了其所包含的值（参见附录C.5）。*function*属性定义了对象的用途或它们的性质，例如玉米地或荒地。而如果对象的实际使用方式与*function*属性定义不同，则可以使用*usage*属性。这两个属性都可以出现多次。

*LandUse*对象在LOD 0-4所有层级中都具有定义，并且在每一层级的LOD中可能具有不同的几何图形。*LandUse*对象表面的几何图形需要包含三维坐标值。其必须是GML3的*MultiSurface*。它可以指定纹理或颜色等外观属性（此时将使用CityGML的外观模型，参见第9章）。

*LandUse*对象可以用来对地球表面建立连贯的几何/语义细分方式。在这种情况下，通过只定义一次*LineStrings*边界，并使用XLinks在相应的*Polygons*中进行引用，相邻*LandUse*对象之间的拓扑关系就得以落实（参见第8.1章）。图 72展示了土地利用细分。其中土地利用对象的几何图形表示为三角化曲面。事实上，它们是DTM约束三角剖分的结果，并考虑了由土地利用分类的二维矢量地图所定义的分割线。

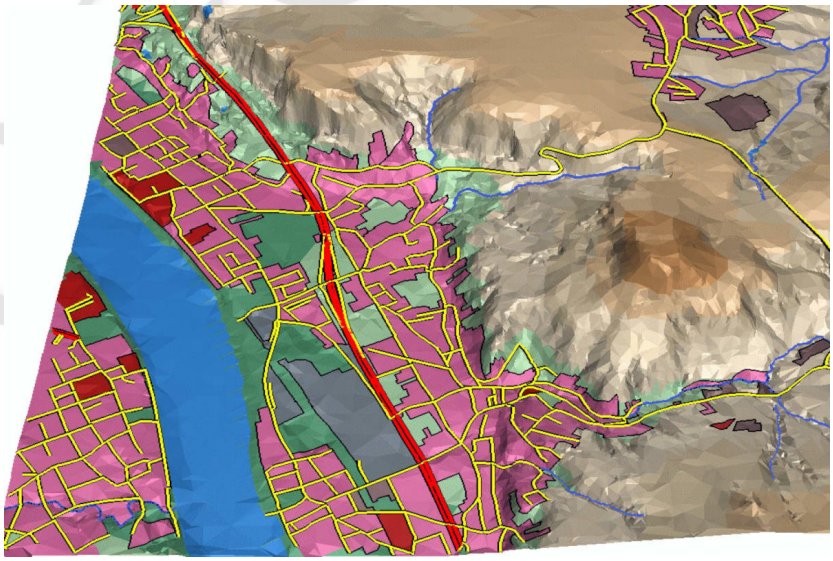

图 72. LOD0的区域模型包括了CityGML的土地利用对象

XML命名空间

CityGML的*LandUse*模块的XML命名空间由统一资源标识符（URI）标识 http://www.opengis.net/citygml/

landuse/2.0. 在*LandUse*模块的XML模式定义中，此URI还用于标识默认的命名空间。

10.10.1. 土地使用对象

LandUseType, LandUse

```
<xs:complexType name="LandUseType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod0MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod1MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod2MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod3MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element name="lod4MultiSurface" type=
"gml:MultiSurfacePropertyType" minOccurs="0"/>
                <xs:element ref="_GenericApplicationPropertyOfLandUse"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
=========================================================================
============== -->
<xs:element name="LandUse" type="LandUseType" substitutionGroup=
"core:_CityObject"/>
<!--
=========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfLandUse" type="xs:anyType"
abstract="true"/>
```

10.10.2. 代码列表

LandUse的*class*，*function*和*usage*属性指定为*gml:CodeType*。这些属性的值将在代码列表中进行列举。相应的代码列表可见附录C.5。

10.10.3. 一致性要求

基础要求

*gml:MultiSurface*几何元素在表示*LandUse*对象的几何图形时，必须在每个LOD层级提供三维坐标值。

10.11. 城市对象组

第6.8章介绍了CityGML对象组的概念。*CityObjectGroup*使用软件工程中的复合设计模式进行建模（参见Gamma等人，1995）：*CityObjectGroup*聚合了城市对象，并进一步被定义为特殊的*城市对象*。这意味着一个组可能成为另一个组的成员，从而实现了递归的聚合模式。然而，在CityGML的实例文档中，必须（通过生成模型的应用程序）确保其不包含循环分组。图 73显示了*CityObjectGroup*的UML图。其XML模式可参见附录A.6。CityGML的分组概念由专题扩展模块 *_CityObject*Group__定义（参见第7章）。

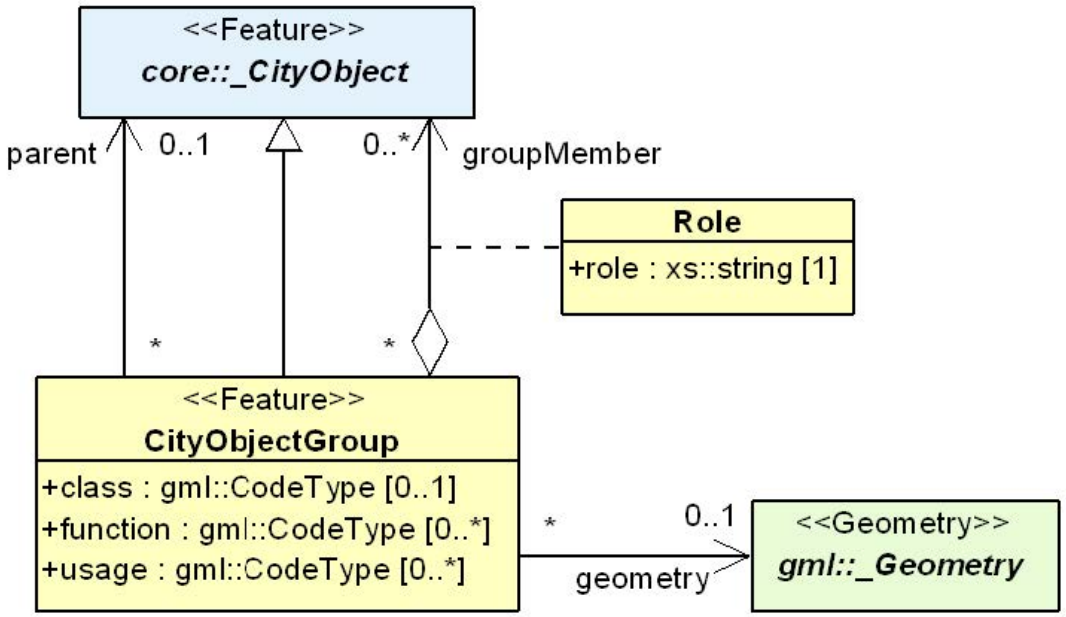

图 73. CityGML城市对象组的UML图。前缀用于表示与模型元素相关的XML命名空间。没有前缀的元素名字在CityGML的 _CityObjectGroup__模块中定义。

*CityObjectGroup*具有*class*，*function*和*usage*属性。*class*属性允许对其所描述的功能进行组分类，并且只允许出现一次。*function*属性用来表示对象组的主要功能，有可能是它所属的专题领域（例如场地，建筑，交通，建筑，未知等）。如果对象的实际使用方式与*function*不同，则可以使用*usage*属性。这两个属性都可以出现多次。对象组中的每个成员都可以通过*role*名字来进行特殊标定，以反映每个_ _*CityObject*在对象组中所扮演的角色。此外，*CityObjectGroup*可以从第8.1章图9所示的GML3子集中选择性地选取几何体对象，用于表示该对象组成员的几何体所生成的通用几何体。

将 *CityObjectGroup对象链接到* _CityObject__对象的父关联使得采用通用层次分组概念进行建模成为可能。被命名的组件组合体（即城市对象）可以被添加到某一被视为父对象的特定城市对象。父关联与组合对象相连，同时其组成员提供了部件信息。该概念可用于表示建筑物中的楼层（参见第10.3.6节：使用城市对象组对建筑物楼层进行建模）。

XML命名空间

CityGML *_CityObject*Group*模块的XML命名空间由统一资源标识符（URI）标识 http://www.opengis.net/citygml/* _CityObject*group/2.0. 在* _CityObject*Group*模块的XML模式定义中，此URI还用于标识默认命名空

间。

10.11.1. 城市对象组

_CityObjectGroupType, _CityObjectGroup

```
<xs:complexType name="CityObjectGroupType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="groupMember" type=
"CityObjectGroupMemberType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="parent" type="CityObjectGroupParentType"
minOccurs="0"/>
                <xs:element name="geometry" type="
gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element ref=
"_GenericApplicationPropertyOfCityObjectGroup" minOccurs="0" maxOccurs=
"unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
=========================================================================
============== -->
<xs:element name="CityObjectGroup" type="CityObjectGroupType"
substitutionGroup="core:_CityObject"/>
<!--
=========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityObjectGroup" type=
"xs:anyType" abstract="true"/>
<!--
=========================================================================
============== -->
<xs:complexType name="CityObjectGroupMemberType">
    <xs:sequence minOccurs="0">
        <xs:element ref="core:_CityObject"/>
    </xs:sequence>
    <xs:attribute name="role" type="xs:string"/>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

```
<!--
================================================================================
============== -->
<xs:complexType name="CityObjectGroupParentType">
    <xs:sequence minOccurs="0">
        <xs:element ref="core:_CityObject"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

10.11.2. 代码列表

*CityObjectGroup要素的class，function和usage属性指定为gml:CodeType_*。这些属性的值将在代码列表中进行列举。相应的代码列表可参见附录C.10。

10.11.3. 一致性要求

基础要求

1. CityGML实例文档中不得包含循环分组。

引用一致性

2. *CityObjectGroup元素的groupMember属性（类型： _CityObjectGroupMemberType）可以在内部包含一个core: CityObject元素，或借助GML 3.1.1的XLink概念，包含一个对外部core: CityObject元素的XLink引用。在后一种情况下，groupMember属性的XLink:href属性只能指向外部的core: CityObject元素（其中外部的core: _CityObject*元素位于另一份文档中，或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。
3. *CityObjectGroup的parent（类型： _CityObjectGroupParentType）属性可以包含在内部定义过的core: CityObject元素或使用XLink对外部的core: CityObject元素进行引用。在后一种情况下parent属性的xlink:href 属性只能指向外部的core: CityObject元素（其中外部core: _CityObject*元素位于另一个文档或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.12. 泛型城市对象与属性

泛型城市对象和属性的概念允许存储和交换未被CityGML经过明确定义的任何专题类涵盖的三维对象，或包含CityGML尚未表达属性的三维对象。这些对CityGML数据模型的泛型扩展，是通过在专题扩展模块*Generics*中定义的*Generic* _CityObject*类和*_genericAttribute__类实现的（参见第7章）。为了避免语义互用性方面的问题，只有在任何其他CityGML模块都未能提供合适的专题类或属性时，才使用泛型扩展。

图 74显示了泛型对象和属性的UML图。XML模式定义见下文和附录A.7。

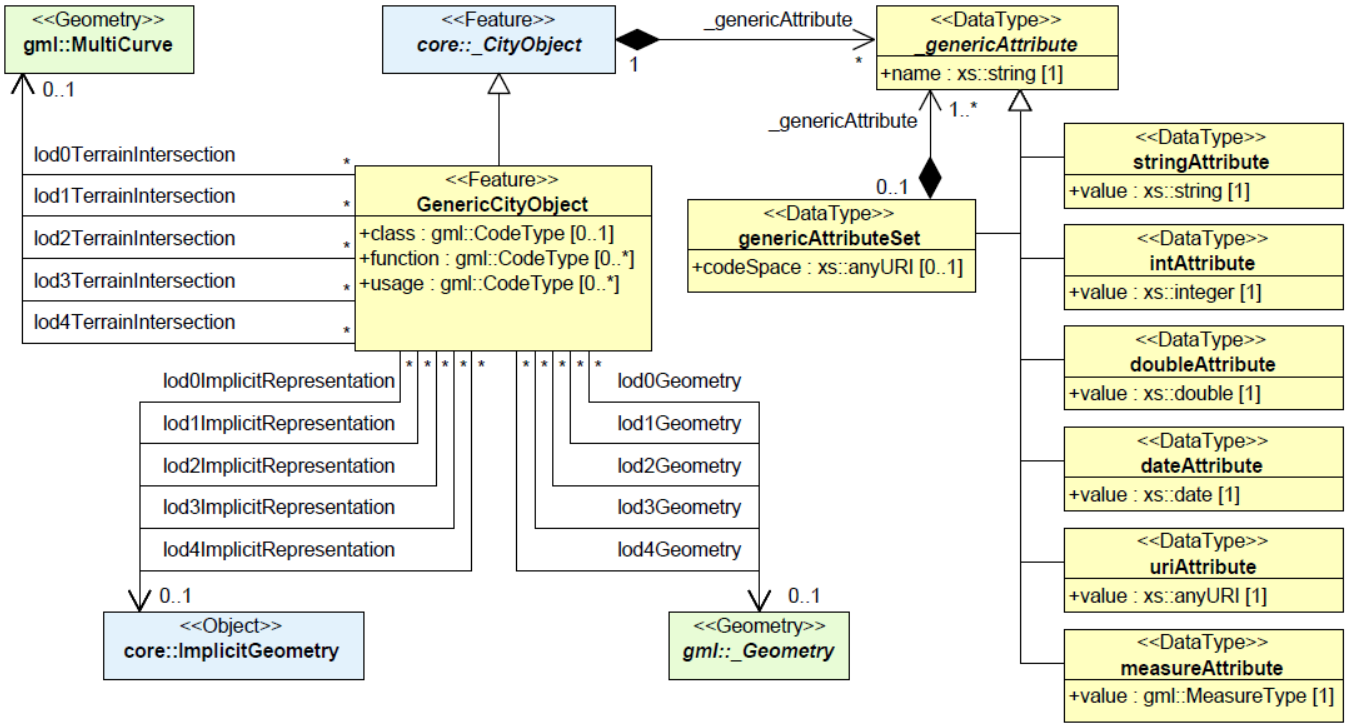

图 74. CityGML泛型对象和属性的UML图。前缀用于表示与模型元素相关的XML命名空间。无前缀的的元素名字将在CityGML的Generics模块中定义。

泛型属性是与城市对象关联的“名称-数值”对。每个泛型属性都有一个可自由选择但必须存在的标识符*name*。属性值的数据类型可以是*String*，*Integer*，*Double*（浮点数），*URI*，*Date*和*gml:MeasureType*。属性类型通过选择*_genericAttribute*的特定子类来定义，例如*stringAttribute*，*intAttribute*等。*measureAttribute*有助于表达测量值。它的值属于结构化类型*gml:MeasureType*。该类型提供*xs:anyURI*类型的可选属性*uom*(units of measure，测量单位)。其中*xs:anyURI*类型指向着表达该单位大小的参考系统。

泛型属性可以通过使用*genericAttributeSet*被分配到拥有各自统一名称的不同组之下。*genericAttributeSet*类继承了*genericAttribute*类，因此它也以泛型属性的形式呈现。它的值是被包含的泛型属性的集合。它的*name*属性为整个集合提供了名称标识符。由于*genericAttributeSet*本身就是泛型属性，它也可以被包含在泛型属性集合之中，从而实现了任意深度的递归嵌套。*genericAttributeSet*的*codeSpace*属性（属于*xs:anyURI*类型）用于将属性集合与某一主体相关联，例如，定义了该属性集及其所包含属性的组织或社会团体。通过这种方法，即使泛型属性集享有相同的名称，它们也可以被清楚地区分开来。

为了对泛型属性建模，CityGML*Core*模块中定义的抽象基类*_ CityObject通过使用CityGML的应用领域扩展机制的附加属性元素 _genericAttribute进行了自我扩充（参见第6.12章）。通过这种方式， CityObject的每个专题子类继承了该属性，并因此可以被分配到任意数量的泛型属性。 这样就可以对CityGML数据模型已定义好的专题类中没有表达出来的要素属性进行表达。因此，泛型模块对定义 _CityObject*之下专题子类的所有CityGML扩展模块都有一定的影响。

Generic _CityObject*的*class，function*和*usage*属性可以定义为*gml:CodeType。class*属性允许对专题领域内的对象进行分类，例如管道，电力线，水坝或未知对象。*function*属性描述该*Generic *_CityObject*所属的专题领域（例如，场地，交通，建筑，能源供应，供水，未知等）。如果对象的实际使用方式与*function*不同，则可以使用*usage*属性。这两个属性都可以出现多次出现。

Generic _CityObject*的几何体既可以是显式的GML3几何体，也可以是ImplicitGeometry（见第8.2章）。在显式几何体的情况下，对于每个LOD层级，对象只能有一个几何体。该几何体可以是任意的三维GML几何体对象（*

*即*gml:_Geometry*类。它是所有GML几何体*lodXGeometry*的基类，其中X∈[0..4])。对于显式几何体，必须要根据城市模型的坐标系给出相应的绝对坐标值。对于ImplicitGeometry，则必须要给出对象的参考点（锚点），以及可选的变换矩阵。为了计算出对象的实际位置，必须将局部坐标转换为城市模型的坐标系，并且必须添加锚点坐标。ImplicitGeometry的形状可以以具有专有格式的外部资源形式给出，例如来自本地文件系统或外部Web服务的VRML或DXF文件。此外，可以使用*relativeGeometry__属性，把形状定义为具有局部笛卡尔坐标的三维GML3几何体（更多详细信息请参见第8.2章）。

为了准确描述DTM与*Generic* _CityObject*的三维几何体的交集，后者可以在每个LOD层级包含*TerrainIntersectionCurves*（参见第6.5章）。这对于三维可视化，以及驾驶模拟器等特定应用而言很重要。例如，如果一座城墙（例如中国的长城）被表示为*Generic *_CityObject*，则必须确保DTM和城墙之间的平稳过渡。

XML命名空间

CityGML的*泛型*模块的XML命名空间由统一资源标识符（URI）标识 http://www.opengis.net/citygml/generics/2.0. 在*泛型*模块的XML模式定义中，该URI还用于标识默认命名空间。

10.12.1. 泛型城市对象

Generic _CityObjectType, Generic _CityObject

```
<xs:complexType name="GenericCityObjectType">
    <xs:complexContent>
        <xs:extension base="core:AbstractCityObjectType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType" minOccurs="0
"/>
                <xs:element name="function" type="gml:CodeType" minOccurs
="0" maxOccurs="unbounded"/>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="lod0Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod1Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod2Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod3Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod4Geometry" type=
"gml:GeometryPropertyType" minOccurs="0"/>
                <xs:element name="lod0TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod4TerrainIntersection" type=
```

```
"gml:MultiCurvePropertyType" minOccurs="0"/>
                <xs:element name="lod0ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
                <xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="GenericCityObject" type="GenericCityObjectType"
substitutionGroup="core:_CityObject"/>
```

10.12.2. 泛型属性

AbstractGenericAttributeType, _genericAttribute, StringAttributeType, stringAttribute等

```
<xs:complexType name="AbstractGenericAttributeType" abstract="true">
    <xs:sequence/>
    <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_genericAttribute" type="AbstractGenericAttributeType"
abstract="true" substitutionGroup=
"core:_GenericApplicationPropertyOfCityObject"/>
<!--
================================================================================
=============== -->
<xs:complexType name="StringAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="xs:string"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
```

```
================================================================================
=============== -->
<xs:element name="stringAttribute" type="StringAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
=============== -->
<xs:complexType name="IntAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="xs:integer"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="intAttribute" type="IntAttributeType" substitutionGroup
="_genericAttribute"/>
<!--
================================================================================
=============== -->
<xs:complexType name="DoubleAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="xs:double"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="doubleAttribute" type="DoubleAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
=============== -->
<xs:complexType name="DateAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="xs:date"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
```

```
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="dateAttribute" type="DateAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
=============== -->
<xs:complexType name="UriAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="xs:anyURI"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="uriAttribute" type="UriAttributeType" substitutionGroup
="_genericAttribute"/>
<!--
================================================================================
=============== -->
<xs:complexType name="MeasureAttributeType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element name="value" type="gml:MeasureType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="measureAttribute" type="MeasureAttributeType"
substitutionGroup="_genericAttribute"/>
```

GenericAttributeSetType, genericAttributeSet

```
<xs:complexType name="GenericAttributeSetType">
    <xs:complexContent>
        <xs:extension base="AbstractGenericAttributeType">
            <xs:sequence>
                <xs:element ref="_genericAttribute" minOccurs="1"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="codeSpace" type="xs:anyURI" use="optional
"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="genericAttributeSet" type="GenericAttributeSetType"
substitutionGroup="_genericAttribute"/>
```

10.12.3. 代码列表

*Generic CityObject要素的class，function和usage属性可以被指定为gml:CodeType_*。这些属性的值将在代码列表中进行列举。

10.12.4. 一致性要求

泛型城市对象和属性的使用限制

1. *Generic* _CityObject__元素只应当被用于对CityGML其他模块都没有提供的专题城市对象——也就是CityGML整体数据模型没有涵盖的对象进行建模。如果有合适的专题类可用，则应使用该专题类，且相应的CityGML模块也必须被CityGML实例文档采用。

2. *core:*_ _CityObject*的*genericAttribute__属性只能用于描述CityGML数据模型专门定义的专题类中未能表达的要素的属性。因此，只有当表达该要素的合适的专题类没法提供合适属性的情况下，才应当使用泛型属性进行建模。

引用一致性

3. *Generic* _CityObject*元素的*lodXImplicitRepresentation*属性，X∈[0..4]（所属类型*：core:ImplicitRepresentationPropertyType）*可以在内部包含*core:ImplicitGeometry*元素，或借助GML 3.1.1的XLink概念，包含对外部的*core:ImplicitGeometry*元素的XLink引用。在后一种情况下*，lodXImplicitRepresentation*属性，X∈[0..4]的*XLink:href*属性只能指向外部的*core:ImplicitGeometry*元素（其中外部的*core:ImplicitGeometry__元素位于另一份文档中，或同一文档中的其他位置）。被包含的元素及对外部的引用，只需提供其中之一，而不能两者都提供，或两者都不提供。

10.13. 应用领域扩展（ADE）

CityGML被设计为独立于应用的信息模型，并负责为三维城市和景观模型转换格式。但是，部分特定的应用通常

需要对其他类型的信息进行建模和格式转换。通常来说，有两种不同的方法把城市模型数据和应用数据进行结合：

1. 将CityGML对象嵌入到（更大的）应用框架中，并在应用框架内建立起应用数据和CityGML数据之间的联系。例如，CityGML数据片段可以基于应用的数据模型结构被嵌入到应用的XML数据文件中，或存储为应用对象的属性。
2. 将与应用直接相关的信息容纳到CityGML实例文档中。该方法是可行的，尤其是当应用的这些信息的组织遵循着与CityGML模式定义相同的结构的时候。如果应用数据可以使用CityGML对象的附加属性进行表达，这种方法就进一步可行了。此时只需要再定义新的要素类型。

在下文中，我们将关注第二种方式，因为只有这种方法处于本规范的涉及领域内。泛型属性和对象被认为是最有可能支持应用的特定数据格式转换（参见第10.12节）。虽然已经可以实现在不更改XML模式定义的情况下扩展CityGML，但是这种灵活性有以下缺点：

- 泛型属性和对象可以在CityGML实例文档中随意出现，但是目前没有对名称，数据类型和多重性进行正式规范。因此，对于应用而言，不能保证泛型属性的实例在每个CityGML要素中将会被包含的最少或最多的次数的。与预定义的CityGML对象不同，泛型对象和属性的具体排布方式和出现次数无法通过XML解析器进行验证。这可能会降低语义的互用性。
- 如果CityGML实例文档需要同时扩充来自不同应用的特定信息，则可能发生泛型属性或对象的命名冲突。
- 只有部分的预定义数据类型可被用于泛型属性。此外，泛型对象的结构可能不适合表达更复杂的对象。

如果应用信息的结构良好，则最好以系统的方式来表达它们，即，基于CityGML的模式定义对额外的正式模式进行定义。这种XML模式称为CityGML应用领域扩展（ADE）。它允许基于经过扩展的CityGML和ADE模式对实例文档进行验证，因此有助于维护在同一应用领域工作的不同系统之间的语义和句法的互用性。为了防止命名冲突，每个ADE都必须在其所属的命名空间中进行定义。该命名空间必须和与CityGML模块相关联的命名空间不同。一个ADE模式可扩展一个或多个CityGML模块模式。相关的CityGML模块模式必须通过ADE模式导入。

ADE概念定义了一种扩展现有CityGML要素类型的特殊方式。其允许在同一实例文档中同时使用不同的ADE（见下文）。例如，ADE的规范可用于以下应用领域：文化遗产（抽象类 _ _*CityObject*的扩展，例如，时间阶段信息和纪念碑保护状态）；地下物体（隧道，地下通道）或城市照明（路灯和房屋灯光等光源）的表达；房地产管理（CityGML要素的经济参数；包含由OSCRE定义的为房地产资产定义的属性）；公用事业网络（作为地形特征）；美国国家建筑信息模型标准（NBIMS）定义的其他建筑属性。

10.13.1. ADE技术原理

每个ADE由其自己的XML模式文件进行定义。其目标命名空间由指定CityGML ADE的信息社区提供。该社区通常不是OGC，SIG 3D。该命名空间应受其所属的信息社区管辖，并且必须以未曾使用且全局唯一的URI的形式给出。此URI将在CityGML ADE实例文档中用于使其区别于CityGML的基础元素。由于URI指代着信息社区，它同时也表示该ADE的发起者。

ADE的XML模式文件应可被所有需要创建和解析CityGML实例文档（包括ADE特定的扩展）的人获取，或者在互联网上能被访问。

ADE的XML模式可以定义CityGML的各种扩展。但是，所有扩展都应属于以下两个类别之一：

1. 新的要素类型需要在ADE命名空间中定义，并且是基于CityGML的抽象或具体类。通常来说，此机制遵循与GML应用模式定义相同的原则。这意味着新的要素类型必须从现有（此处即指CityGML）要素类型继承而来。例如，新的要素类型可以通过诸如 *_CityObject*或*AbstractBuilding*的抽象类，或诸如*CityFurniture*的具体类的子类来定义。新的要素类型自动从各自的CityGML父类中继承所有的属性和关联。

2. 现有CityGML要素类型可扩展至应用特定的属性（在ADE命名空间中）。这些属性可以具有简单或复杂的数据类型，也可包括几何信息或嵌套于其中的要素（即要素属性）。后者还可用于对和其他要素的关系进行建模。

   在这种情况下，CityGML的要素类型的扩展并不是通过XML模式的继承机制来实现的。相反，每个CityGML要素类型在其XML模式定义中都提供了一个“钩”（hook），以允许ADE为其附加其他属性。这个“钩”是形式为*GenericApplicationPropertyOf<要素类型名称>*的GML属性，其中*<要素类型名称>*和包含它的要素类型的定义名称相同。这些类型的属性的数据类型都来源于XSD命名空间的*xsd:anyType*。其中*GenericApplicationPropertyOf<要素类型名称>*的最小出现次数为0，最大出现次数则无限制。这意味着，CityGML模式允许每个CityGML要素可以具有任意数量的附加属性，其中包含*_GenericApplicationPropertyOf<要素类型名称>*的任意XML内容。例如，CityGML要素类型*LandUse*定义中的最后一个属性是*GenericApplicationPropertyOfLandUse*元素（参见第10.10.1章）。

   此类属性被称为“钩”，表示可附加特定的应用属性，因为它们被ADE用作替代组的头部。每当ADE想要向现有CityGML要素类型添加额外属性的时候，它应在ADE命名空间中使用合适的数据类型声明相应的元素。在元素声明中，该元素应明确被分配的相应的替代组。该替代组由对应的CityGML模块的命名空间中对应的*GenericApplicationPropertyOf<要素类型名称>*进行定义。后续小节将给出一个示例。

   通过遵循上述概念，即可以为不同的信息社区指定不同的ADE。每个ADE都可以将它们特定的属性添加到相同的CityGML要素类型中，因为它们都可属于同一个替代组。此时，CityGML实例文档便可让CityGML要素同时包含不同ADE的附加信息。

需要注意的是，使用ADE会带来额外的复杂性，因为数据文件可能包含来自不同命名空间的混杂在一起的信息（包括要素，属性），而不仅仅来自GML和CityGML模块命名空间。然而，被扩展的CityGML实例文档很容易会被非“模式意识”的应用处理，即不以通用的方式解析和编译GML应用模式的应用。这些应用可能会简单地跳过CityGML实例文档中任何不是来自CityGML模块或GML命名空间的内容。因此，建筑物仍需要由具有标准CityGML属性的<bldg:Building>元素进行表达，但仍可能会包含来自不同命名空间的属性。同样，来自不同于CityGML模块或GML的命名空间的要素也会被跳过（例如，可视化应用）。

10.13.2. ADE示例

在本节中，ADE的机制将通过一个简短的示例进行说明。该示例涉及虚拟三维城市模型应用于生成噪声污染地图。在我们的示例中，此任务需要CityGML的两个扩展：建筑物需要被扩展，以便表达“噪声反射校正”值和居民数量；作为一种新的要素类型，必须对噪声屏障进行定义，且该类型也具有“噪声反射校正”值。

为了实现该模型而定义的XSD模式为噪声扩展声明了一个新的命名空间（ http://www.citygml.org/ade/noise_de/2.0 ）。此外，还声明了CityGML扩展模块的命名空间（相应的前缀请参见第4.3章和第7章），并导入相应的模式定义文件。XML模式添加了*buildingReflectionCorrection*和*buildingHabitants*元素。这两个元素都是由CityGML*Building*模块定义的替换组*bldg:_GenericApplicationPropertyOfAbstractBuilding*的成员。因此，这两个元素都可以用作CityGML建筑要素的子元素。噪声屏障则表示为*NoiseCityFurnitureSegment*元素。对应的类型*NoiseCityFurnitureSegmentType*被定义为CityGML*Core*模块提供的CityGML抽象类型*core:Abstract* _CityObject*Type*的子类型，并被赋予了XML和XSD的分级机制。另一个元素*noiseCityFurnitureSegmentProperty*也被添加到替代组*frn:_GenericApplicationPropertyOfCityFurniture*

作为其中的成员。通过这种方式，隔音屏障可以按照CityGML城市家具对象的子元素的形式进行建模。

CityGML噪声ADE的这个示例的XSD文件如以下片段所示（完整的CityGML噪声ADE可在附录H中查看）：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.citygml.org/ade/noise_de/2.0"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:core="http://www.opengis.net/citygml/2.0"
    xmlns:bldg="http://www.opengis.net/citygml/building/2.0"
    xmlns:frn="http://www.opengis.net/citygml/cityfurniture/2.0"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"targetNamespace=
"http://www.citygml.org/ade/noise_de/2.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xsd:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
    <xsd:import namespace="http://www.opengis.net/citygml/2.0"
schemaLocation="http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
    <xsd:import namespace=
"http://www.opengis.net/citygml/transportation/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/transportation/2.0/transportation.xsd"
/>
    <xsd:import namespace="http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://schemas.opengis.net/citygml/building/2.0/building.x
sd"/>
    <xsd:import namespace=
"http://www.opengis.net/citygml/cityfurniture/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/cityfurniture/2.0/cityFurniture.xsd"/>
 …
    <xsd:element name="buildingReflection" type="xsd:string"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
    <xsd:element name="buildingHabitants" type="xsd:positiveInteger"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
 …
    <xsd:element name="noiseCityFurnitureSegmentProperty" type=
"NoiseCityFurnitureSegmentPropertyType" substitutionGroup=
"frn:_GenericApplicationPropertyOfCityFurniture"/>
    <!--
================================================================================
================= -->
    <xsd:complexType name="NoiseCityFurnitureSegmentPropertyType">
        <xsd:sequence minOccurs="0">
            <xsd:element ref="NoiseCityFurnitureSegment" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:complexType>
    <!--
================================================================================
================= -->
    <xsd:complexType name="NoiseCityFurnitureSegmentType">
```

```
        <xsd:complexContent>
            <xsd:extension base="core:AbstractCityObjectType">
                <xsd:sequence>
                    <xsd:element name="type" type=
"NoiseCityFurnitureSegmentTypeType" minOccurs="0"/>
                    <xsd:element name="reflection" type="xsd:string"
minOccurs="0"/>
                    <xsd:element name="reflectionCorrection" type=
"gml:MeasureType" minOccurs="0"/>
                    <xsd:element name="height" type="gml:LengthType"
minOccurs="0"/>
                    <xsd:element name="distance" type="gml:LengthType"
minOccurs="0"/>
                    <xsd:element name="lod0BaseLine" type=
"gml:CurvePropertyType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!--
================================================================================
================== -->
    <xsd:element name="NoiseCityFurnitureSegment" type=
"NoiseCityFurnitureSegmentType" substitutionGroup="core:_CityObject"/>
 …
 </xsd:schema>
```

下面描述了相应的实例文档中要素集合的示例。两个CityGML建筑物包含的应用属性，通过在命名空间的前缀*noise*实现与CityGML属性的区分。其他属性，*function*和*geometry*，由相应的CityGML模块定义。除了建筑物之外，作为城市家具元素的子元素的隔音屏障，也被包含在要素集合中。需要注意的是，序列中子元素的顺序并非是任意的：ADE子模式定义的子元素必须出现在CityGML模块定义的子元素之后。但是ADE的属性中没有规定排序。

```
…
<core:cityObjectMember>
    <bldg:Building gml:id="aa">
        <bldg:function>1004</bldg:function>
        <bldg:lod1Solid> ... </bldg:lod1Solid>
        <noise:buildingHabitants>14</noise:buildingHabitants>
        <noise:buildingReflectionCorrection uom="dB">
4.123</noise:buildingReflectionCorrection>
    </bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
    <bldg:Building gml:id="aaa">
        <bldg:function>1004</bldg:function>
        <bldg:lod1Solid> ... </bldg:lod1Solid>
        <noise:buildingReflectionCorrection uom="dB">
3.123</noise:buildingReflectionCorrection>
        <noise:buildingHabitants>6</noise:buildingHabitants>
    </bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
    <frn:CityFurniture gml:id="CFUR_0815">
        <frn:function>1520</frn:function>
        <frn:lod1Geometry> … </frn:lod1Geometry>
        <noise:noiseCityFurnitureSegmentProperty>
            <noise:NoiseCityFurnitureSegment gml:id="CFRS_0815">
                <noise:type>1</noise:type>
                <noise:reflection>absorbierende
Lärmschutzwand</noise:reflection>
                <noise:reflectionCorrection uom="dB">
4.123</noise:reflectionCorrection>
                <noise:height uom="m">7.123</noise:height>
                <noise:distance uom="m">21.123</noise:distance>
                <noise:lod0BaseLine>
                    <gml:LineString srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783" srsDimension="3">
                        <gml:coordinates decimal="." cs="," ts=" "
>5707335,2524175,188 5707338,2524181,188 5707330,2524185,188
5707327,2524179,188</gml:coordinates>
                    </gml:LineString>
                </noise:lod0BaseLine>
            </noise:NoiseCityFurnitureSegment>
        </noise:noiseCityFurnitureSegmentProperty>
    </frn:CityFurniture>
</core:cityObjectMember>
…
```

10.14. 代码列表

CityGML要素类型一般会包括可以通过离散值进行枚举的属性。例如，建筑物的*roof type*属性就是这样的一个示例。其属性值通常包括马鞍形屋顶，四坡屋顶，半坡屋顶，平屋顶，斜屋顶，或帐篷屋顶。如果此类属性直接通过字符串输入，拼写错误，同一概念的不同名称将会破坏互用性。

如果值列表是固定的，其将在CityGML模式中通过使用*enumeration*属性类型指定其所允许的属性值并强制执行。枚举类型的属性只能从预定义列表中获取值。这样的属性的示例包括抽象基类*core:*_ _CityObject（*CityGML*Core*模块，参见第10.1章）的*relativeToTerrain*和*relativeToWater*属性，以及抽象类*app:_Texture（Apparence*模块，参见第9.4章）的*wrapMode__属性。

如果枚举所有可能出现的属性值的方法不合适，则需要将属性类型指定为*gml:CodeType*。其允许的属性值可通过在CityGML模式之外定义的*代码列表*提供。此类属性的例子包括*class*，*function*和*usage*。它们可用于几乎所有CityGML要素类型。代码列表包含了防止拼写错误的经过编码的属性值，并确保相同的编码被用于相同的概念。如果为*gml:CodeType*类型的属性提供代码列表，则只能从给定的代码列表中获取数值。这允许应用去验证属性值，从而促进了语义和句法的互用性。为*gml:CodeType*声明的*codeSpace*可选属性用于将属性与相应的代码列表建立联系。如果存在这样的*codeSpace*，则其值应为标识代码列表的恒定URI。如果没有给出*codeSpace*，那么属性值只能被编译为简单的文本标记，并需要额外的知识对其进行验证。

代码列表的治理与CityGML模式和规范的治理相互分离。因此，任何组织或信息社区都可以根据他们的信息需求来指定代码列表。每个*codeSpace*应只有一个治理主体，因此每个代码列表同样也只有一个治理主体，对代码列表的内容和维护工作负责。此时，代码列表值在CityGML模式之外被管理。因此，与CityGML模式强制执行的固定枚举的方法相比，代码列表的更改不需要对CityGML模式和规范进行修订。

对于不同的国家代码列表（例如，源于国家法律或法规）和不同的信息社区而言，代码列表的内容可能具有很大差异。为此，该国际标准没有为*gml:CodeType*类型的任何属性指定规范的代码列表。但是，附录C为由SIG 3D提出和维护的部分属性提供了非规范性的代码列表。这些代码列表可以在CityGML实例文档中直接引用，并作为代码列表定义的示例。附录C中给出的代码列表包括非规范性的代码列表。这些代码列表被收录在本国际标准的1.0旧版本中，以确保向后兼容性。

推荐按照GML 3.1.1的*Simple Dictionary Profile*（参见Whiteside，2005）将代码列表以*simple dictionary*的形式实现。下面给出了以*simple dictionary*形式实现的代码列表的示例。它展示了SIG 3D提出的代码列表的一个片段。其为*_AbstractBuilding*类（*Building*模块，参见第10.3章）的*roofType*属性的代码列表。

```
<gml:Dictionary xmlns:gml="http://www.opengis.net/gml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/gml
http://schemas.opengis.net/gml/3.1.1/profiles/SimpleDictionary/1.0.0/gmlSi
mpleDictionaryProfile.xsd"
    gml:id="roofType">
    <gml:name>roofType</gml:name>
    <gml:dictionaryEntry>
        <gml:Definition gml:id="id357">
            <gml:description>flat roof</gml:description>
            <gml:name>1000</gml:name>
        </gml:Definition>
    </gml:dictionaryEntry>
    <gml:dictionaryEntry>
        <gml:Definition gml:id="id358">
            <gml:description>monopitch roof</gml:description>
            <gml:name>1010</gml:name>
        </gml:Definition>
    </gml:dictionaryEntry>
    …
</gml:Dictionary>
```

在简单字典的概念中，代码列表本身由*gml:Dictionary*元素表示。其允许的属性取值由*gml:Dictionary*所包含的*gml:Definition*条目列出。对于每一项定义，被编码的属性值都通过*gml:name*子元素来指定。CityGML实例文档中任何引用该代码列表的属性，只能使用定义条目的*gml:name*元素所指定的属性值中选取。如果属性值未被任意一项定义条目指定，则该属性值无效。定义条目中的*gml:description*子元素为被编码的属性值提供了额外的文本描述。例如，该描述可用作对被编码的属性值的易读的注释。

以下CityGML实例文档的摘录说明了代码列表机制的用法。该文档包含一个*bldg:Building*对象。其*roofType*值取自列表2所示的代码列表。*roofType*元素的*codeSpace*属性通过全局唯一的由SIG 3D进行管理和维护的URL http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_roofType.xml 对代码列表进行定义。根据此代码列表，编码属性值*1000*表示该建筑物的*flat roof*。

```
<bldg:Building>
    <bldg:roofType codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_ro
ofType.xml">1000</bldg:roofType>
    …
</bldg:Building>
```

Annex A: （规范性）XML模式定义

A.1. CityGML核心模块

*CityGML Core*模块在XML模式定义文件*cityGMLBase.xsd*中被定义。其目标命名空间 http://www.opengis.net/citygml/2.0 和核心模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/2.0" elementFormDefault=
"qualified" attributeFormDefault="unqualified" version="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
schemaLocation="http://docs.oasis-open.org/election/external/xAL.xsd"/>
<xs:complexType name="CityModelType">
<xs:annotation>
<xs:documentation>Type describing the "root" element of any city model
file. It is a collection whose members are restricted to be features of a
city model. All features are included as cityObjectMember.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureCollectionType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfCityModel" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="CityModel" type="CityModelType" substitutionGroup=
"gml:_FeatureCollection"/>
<!--
=============================================================================
```

```
=============== -->
<xs:element name="_GenericApplicationPropertyOfCityModel" type="
xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:element name="cityObjectMember" type="gml:FeaturePropertyType"
substitutionGroup="gml:featureMember"/>
<!--
================================================================================
=============== -->
<xs:complexType name="AbstractCityObjectType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the abstract superclass of most CityGML
features. Its purpose is to provide a creation and a termination date as
well as a reference to corresponding objects in other information systems.
A generalization relation may be used to relate features, which represent
the same real-world object in different Levels-of-Detail, i.e. a feature
and its generalized counterpart(s). The direction of this relation is from
the feature to the corresponding generalized feature.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="creationDate" type="xs:date" minOccurs="0"/>
<xs:element name="terminationDate" type="xs:date" minOccurs="0"/>
<xs:element name="externalReference" type="ExternalReferenceType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="generalizesTo" type="GeneralizationRelationType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="relativeToTerrain" type="RelativeToTerrainType"
minOccurs="0"/>
<xs:element name="relativeToWater" type="RelativeToWaterType" minOccurs="
0"/>
<xs:element ref="_GenericApplicationPropertyOfCityObject" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_CityObject" type="AbstractCityObjectType" abstract=
"true" substitutionGroup="gml:_Feature"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfCityObject" type=
```

```
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="AbstractSiteType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the abstract superclass for buildings,
facilities, etc. Future extensions of CityGML like bridges and tunnels
would be modelled as subclasses of _Site. As subclass of _CityObject, a
_Site inherits all attributes and relations, in particular an id, names,
external references, and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractCityObjectType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfSite" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_Site" type="AbstractSiteType" abstract="true"
substitutionGroup="_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfSite" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="GeneralizationRelationType">
<xs:annotation>
<xs:documentation>Denotes the relation of a _CityObject to its
corresponding _CityObject in higher LOD, i.e. to the _CityObjects
representing the same real world object in higher LOD. The
GeneralizationRelationType element must either carry a reference to a
_CityObject object or contain a _CityObject object inline, but neither
both nor none.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_CityObject"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

```
<!--
=======================================================================
=============== -->
<xs:complexType name="ExternalReferenceType">
<xs:annotation>
<xs:documentation>Type describing the reference to an corresponding object
in an other information system, for example in the german cadastre ALKIS,
the german topographic information system or ATKIS, or the OS MasterMap.
The reference consists of the name of the external information system,
represented by an URI, and the reference of the external object, given
either by a string or by an URI. If the informationSystem element is
missing in the ExternalReference, the ExternalObjectReference must be an
URI, which contains an indication of the
informationSystem.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="informationSystem" type="xs:anyURI" minOccurs="0"/>
<xs:element name="externalObject" type="ExternalObjectReferenceType"/>
</xs:sequence>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:complexType name="ExternalObjectReferenceType">
<xs:choice>
<xs:element name="name" type="xs:string"/>
<xs:element name="uri" type="xs:anyURI"/>
</xs:choice>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:simpleType name="RelativeToTerrainType">
<xs:annotation>
<xs:documentation>Specifies the spatial relation of a CityObject realativ
to terrain in a qualitative way. The values of this type are defined in
the XML file RelativeToTerrainType.xml, according to the dictionary
concept of GML3.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="entirelyAboveTerrain"/>
<xs:enumeration value="substantiallyAboveTerrain"/>
<xs:enumeration value="substantiallyAboveAndBelowTerrain"/>
<xs:enumeration value="substantiallyBelowTerrain"/>
<xs:enumeration value="entirelyBelowTerrain"/>
</xs:restriction>
</xs:simpleType>
<!--
=======================================================================
```

```
=============== -->
<xs:simpleType name="RelativeToWaterType">
<xs:annotation>
<xs:documentation>Specifies the spatial relation of a CityObject realativ
to the water surface in a qualitative way. The values of this type are
defined in the XML file RelativeToTerrainType.xml, according to the
dictionary concept of GML3.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="entirelyAboveWaterSurface"/>
<xs:enumeration value="substantiallyAboveWaterSurface"/>
<xs:enumeration value="substantiallyAboveAndBelowWaterSurface"/>
<xs:enumeration value="substantiallyBelowWaterSurface"/>
<xs:enumeration value="entirelyBelowWaterSurface"/>
<xs:enumeration value="temporarilyAboveAndBelowWaterSurface"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:complexType name="AddressPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _CityObject to its addresses.
The AddressPropertyType element must either carry a reference to an
Address object or contain an Address object inline, but neither both nor
none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="Address"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="AddressType">
<xs:annotation>
<xs:documentation>Type for addresses. It references the xAL address
standard issued by the OASIS consortium. Please note, that addresses are
modelled as GML features. Every address can be assigned zero or more 2D or
3D point geometries (one gml:MultiPoint geometry) locating the
entrance(s). </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="xalAddress" type="xalAddressPropertyType"/>
<xs:element name="multiPoint" type="gml:MultiPointPropertyType" minOccurs
="0"/>
```

```
<xs:element ref="_GenericApplicationPropertyOfAddress" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="Address" type="AddressType" substitutionGroup=
"gml:_Feature"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAddress" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="xalAddressPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an Address feature to the xAL
address element.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element ref="xAL:AddressDetails"/>
</xs:sequence>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="ImplicitGeometryType">
<xs:annotation>
<xs:documentation> Type for the implicit representation of a geometry. An
implicit geometry is a geometric object, where the shape is stored only
once as a prototypical geometry, e.g. a tree or other vegetation object, a
traffic light or a traffic sign. This prototypic geometry object is re-
used or referenced many times, wherever the corresponding feature occurs
in the 3D city model. Each occurrence is represented by a link to the
prototypic shape geometry (in a local cartesian coordinate system), by a
transforma-tion matrix that is multiplied with each 3D coordinate tuple of
the prototype, and by an anchor point denoting the base point of the
object in the world coordinate reference system. In order to determine the
absolute coordinates of an implicit geometry, the anchor point coordinates
have to be added to the matrix multiplication results. The transformation
matrix accounts for the intended rotation, scaling, and local translation
of the prototype. It is a 4x4 matrix that is multiplied with the prototype
coordinates using homogeneous coordinates, i.e. (x,y,z,1). This way  even
a projection might be modelled by the transformation matrix. The concept
```

```
of implicit geometries is an enhancement of the geometry model of GML3.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType">
<xs:sequence>
<xs:element name="mimeType" type="gml:CodeType" minOccurs="0"/>
<xs:element name="transformationMatrix" type="TransformationMatrix4x4Type"
minOccurs="0"/>
<xs:element name="libraryObject" type="xs:anyURI" minOccurs="0"/>
<xs:element name="relativeGMLGeometry" type="gml:GeometryPropertyType"
minOccurs="0"/>
<xs:element name="referencePoint" type="gml:PointPropertyType"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="ImplicitGeometry" type="ImplicitGeometryType"
substitutionGroup="gml:_GML"/>
<!--
================================================================================
================ -->
<xs:complexType name="ImplicitRepresentationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a _CityObject to its implicit
geometry representation, which is a representation of a geometry by
referencing a prototype and transforming it to its real position in space.
The ImplicitRepresentationPropertyType element must either carry a
reference to a ImplicitGeometry object or contain a ImplicitGeometry
object inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="ImplicitGeometry"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:simpleType name="doubleBetween0and1">
<xs:annotation>
<xs:documentation>Type for values, which are greater or equal than 0 and
less or equal than 1. Used for color encoding, for example.
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:double">
```

```
<xs:minInclusive value="0"/>
<xs:maxInclusive value="1"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:simpleType name="doubleBetween0and1List">
<xs:annotation>
<xs:documentation>List for double values, which are greater or equal than
0 and less or equal than 1. Used for color encoding, for example.
</xs:documentation>
</xs:annotation>
<xs:list itemType="doubleBetween0and1"/>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:simpleType name="TransformationMatrix4x4Type">
<xs:annotation>
<xs:documentation>Used for implicit geometries. The Transformation matrix
is a 4 by 4 matrix, thus it must be a list with 16 items. The order the
matrix element are represented is row-major, i. e. the first 4 elements
represent the first row, the fifth to the eight element the second row,...
</xs:documentation>
</xs:annotation>
<xs:restriction base="gml:doubleList">
<xs:length value="16"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:simpleType name="TransformationMatrix2x2Type">
<xs:annotation>
<xs:documentation>Used for georeferencing. The Transformation matrix is a
2 by 2 matrix, thus it must be a list with 4 items. The order the matrix
element are represented is row-major, i.e. the first 2 elements represent
the first row, the fifth to the eight element the second row,...
</xs:documentation>
</xs:annotation>
<xs:restriction base="gml:doubleList">
<xs:length value="4"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:simpleType name="TransformationMatrix3x4Type">
```

```
<xs:annotation>
<xs:documentation>Used for texture parameterization. The Transformation
matrix is a 3 by 4 matrix, thus it must be a list with 12 items. The order
the matrix element are represented is row-major, i. e. the first 4
elements represent the first  row, the fifth to the eight element the
second row,... </xs:documentation>
</xs:annotation>
<xs:restriction base="gml:doubleList">
<xs:length value="12"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
============== -->
<xs:simpleType name="integerBetween0and4">
<xs:annotation>
<xs:documentation>Type for integer values, which are greater or equal than
0 and less or equal than 4. Used for encoding of the LOD number.
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:integer">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="4"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

A.2. 外观模块

CityGML的*Appearance*模块在XML模式定义文件*appearance.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/appearance/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/appearance/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/appearance/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
```

```
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AppearanceType">
<xs:annotation>
<xs:documentation> Named container for all surface data
(texture/material). All appearances of the same name ("theme") within a
CityGML file are considered a group. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="theme" type="xs:string" minOccurs="0"/>
<xs:element name="surfaceDataMember" type="SurfaceDataPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfAppearance" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="Appearance" type="AppearanceType" substitutionGroup=
"gml:_Feature"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfAppearance" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="AppearancePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a _CityObject to its
appearances. The AppearancePropertyType element must either carry a
reference to a Appearance object or contain a Appearance object inline,
but neither both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="Appearance"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
========================================================================
============== -->
```

```
<xs:element name="appearanceMember" type="gml:FeaturePropertyType"
substitutionGroup="gml:featureMember"/>
<!--
============================================================================
============== -->
<xs:element name="appearance" type="AppearancePropertyType"
substitutionGroup="core:_GenericApplicationPropertyOfCityObject"/>
<!--
============================================================================
============== -->
<xs:complexType name="AbstractSurfaceDataType" abstract="true">
<xs:annotation>
<xs:documentation>Base class for textures and material. Contains only
isFront-flag.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="isFront" type="xs:boolean" default="true" minOccurs="0
"/>
<xs:element ref="_GenericApplicationPropertyOfSurfaceData" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
============== -->
<xs:element name="_SurfaceData" type="AbstractSurfaceDataType" abstract=
"true" substitutionGroup="gml:_Feature"/>
<!--
============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfSurfaceData" type=
"xs:anyType" abstract="true"/>
<!--
============================================================================
============== -->
<xs:complexType name="SurfaceDataPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an Appearance to its surface
data. The SurfaceDataPropertyType element must either carry a reference to
a _SurfaceData object or contain a _SurfaceData object inline, but neither
both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_SurfaceData" minOccurs="0"/>
</xs:sequence>
```

```
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="AbstractTextureType" abstract="true">
<xs:annotation>
<xs:documentation>Base class for textures. "imageURI" can contain any
valid URI from references to a local file to preformatted web service
requests. The linking to geometry and texture parameterization is provided
by derived classes.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractSurfaceDataType">
<xs:sequence>
<xs:element name="imageURI" type="xs:anyURI"/>
<xs:element name="mimeType" type="gml:CodeType" minOccurs="0"/>
<xs:element name="textureType" type="TextureTypeType" minOccurs="0"/>
<xs:element name="wrapMode" type="WrapModeType" minOccurs="0"/>
<xs:element name="borderColor" type="ColorPlusOpacity" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfTexture" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_Texture" type="AbstractTextureType" abstract="true"
substitutionGroup="_SurfaceData"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTexture" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:simpleType name="WrapModeType">
<xs:annotation>
<xs:documentation>Fill mode for a texture. "wrap" repeats the texture,
"clamp" extends the edges of the texture, and "border" fills all undefined
areas with "borderColor"</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="none"/>
<xs:enumeration value="wrap"/>
<xs:enumeration value="mirror"/>
```

```
<xs:enumeration value="clamp"/>
<xs:enumeration value="border"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
=============== -->
<xs:complexType name="ParameterizedTextureType">
<xs:annotation>
<xs:documentation>Specialization for standard 2D textures. "target"
provides the linking to surface geometry. Only gml:MultiSurface and
decendants of gml:AbstractSurfaceType are valid targets. As property of
the link, a texture parameterization either as set of texture coordinates
or transformation matrix is given. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTextureType">
<xs:sequence>
<xs:element name="target" type="TextureAssociationType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfParameterizedTexture"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="ParameterizedTexture" type="ParameterizedTextureType"
substitutionGroup="_Texture"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfParameterizedTexture" type
="xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="GeoreferencedTextureType">
<xs:annotation>
<xs:documentation>Specialization for georeferenced textures, i.e. textures
using a planimetric projection. Such textures contain an implicit
parameterization (either stored within the image file, in an acompanying
world file, or using the "referencePoint" and "orientation"-elements). A
georeference provided by "referencePoint" and "orientation" always takes
precedence. The search order for an external georeference is determined by
the boolean flag preferWorldFile. If this flag is set to true (its default
value), a world file is looked for first and only if it is not found the
```

```
georeference from the image data is used. If preferWorldFile is false, the
world file is used only if no georeference from the image data is
available. The "boundedBy"-property should contain the bounding box of the
projected image data. Since a georeferenced texture has a unique
parameterization, "target" only provides links to surface geometry without
any additional texture parameterization. Only gml:MultiSurface or
decendants of gml:AbstractSurfaceType are valid
targets.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTextureType">
<xs:sequence>
<xs:element name="preferWorldFile" type="xs:boolean" default="true"
minOccurs="0"/>
<xs:element name="referencePoint" type="gml:PointPropertyType" minOccurs=
"0"/>
<xs:element name="orientation" type="core:TransformationMatrix2x2Type"
minOccurs="0"/>
<xs:element name="target" type="xs:anyURI" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfGeoreferencedTexture"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="GeoreferencedTexture" type="GeoreferencedTextureType"
substitutionGroup="_Texture"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfGeoreferencedTexture" type
="xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="TextureAssociationType">
<xs:annotation>
<xs:documentation>Denotes the relation of a texture to a surface, that is
augmented by a TextureParameterization object. The TextureAssociationType
element must either carry a reference to a _TextureParameterization object
or contain a
_TextureParameterization object inline, but neither both nor
none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
```

```
<xs:element ref="_TextureParameterization"/>
</xs:sequence>
<xs:attribute name="uri" type="xs:anyURI" use="required"/>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="AbstractTextureParameterizationType" abstract="true
">
<xs:annotation>
<xs:documentation>Base class for augmenting a link "texture->surface" with
texture parameterization. Subclasses of this class define concrete
parameterizations. Currently, texture coordinates and texture coordinate
generation using a transformation matrix are available.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTextureParameterization"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_TextureParameterization" type=
"AbstractTextureParameterizationType" abstract="true" substitutionGroup=
"gml:_GML"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTextureParameterization"
type="xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="TexCoordListType">
<xs:annotation>
<xs:documentation>Texture parameterization using texture coordinates: Each
gml:LinearRing that is part of the surface requires a separate
"textureCoordinates"-entry with 2 doubles per ring vertex. The "ring"-
attribute provides the gml:id of the target LinearRing. It is prohibited
to link texture coordinates to any other object type than LinearRing.
Thus, surfaces not consisting of LinearRings cannot be textured this way.
Use transformation matrices (see below) or georeferenced textures instead.
```

```
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTextureParameterizationType">
<xs:sequence>
<xs:element name="textureCoordinates" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="gml:doubleList">
<xs:attribute name="ring" type="xs:anyURI" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element ref="_GenericApplicationPropertyOfTexCoordList" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
============== -->
<xs:element name="TexCoordList" type="TexCoordListType" substitutionGroup
="_TextureParameterization"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTexCoordList" type=
"xs:anyType" abstract="true"/>
<!--
=======================================================================
============== -->
<xs:complexType name="TexCoordGenType">
<xs:annotation>
<xs:documentation>Texture parameterization using a transformation matrix.
The transformation matrix "worldToTexture" can be used to derive texture
coordinates from an object's location.This 3x4 matrix T computes the
coordinates (s,t) from a homogeneous world position p as (s,t) = (s'/q',
t'/q') with (s', t', q') = T*p. Thus, perspective projections can be
specified. The SRS can be specified using standard attributes. If an
object is given in a different reference system, it is transformed to the
SRS before applying the transformation. A transformation matrix can be
used for whole surfaces. It is not required to specify it per LinearRing.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTextureParameterizationType">
<xs:sequence>
```

```
<xs:element name="worldToTexture">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="core:TransformationMatrix3x4Type">
<xs:attributeGroup ref="gml:SRSReferenceGroup"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element ref="_GenericApplicationPropertyOfTexCoordGen" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="TexCoordGen" type="TexCoordGenType" substitutionGroup=
"_TextureParameterization"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfTexCoordGen" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="X3DMaterialType">
<xs:annotation>
<xs:documentation>Class for defining constant surface properties. It is
based on X3D's material definition. In addition, "isSmooth" provides a
hint for value interpolation. The link to surface geometry is established
via the "target"-property. Only gml:MultiSurface or decendants of
gml:AbstractSurfaceType are valid targets. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractSurfaceDataType">
<xs:sequence>
<xs:element name="ambientIntensity" type="core:doubleBetween0and1"
default="0.2" minOccurs="0"/>
<xs:element name="diffuseColor" type="Color" default="0.8 0.8 0.8"
minOccurs="0"/>
<xs:element name="emissiveColor" type="Color" default="0.0 0.0 0.0"
minOccurs="0"/>
<xs:element name="specularColor" type="Color" default="1.0 1.0 1.0"
minOccurs="0"/>
<xs:element name="shininess" type="core:doubleBetween0and1" default="0.2"
minOccurs="0"/>
```

```
<xs:element name="transparency" type="core:doubleBetween0and1" default=
"0.0" minOccurs="0"/>
<xs:element name="isSmooth" type="xs:boolean" default="false" minOccurs="
0"/>
<xs:element name="target" type="xs:anyURI" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfX3DMaterial" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="X3DMaterial" type="X3DMaterialType" substitutionGroup=
"_SurfaceData"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfX3DMaterial" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:simpleType name="TextureTypeType">
<xs:annotation>
<xs:documentation>Textures can be qualified by the attribute textureType.
The textureType differentiates between textures, which are specific for a
certain object and are only used for that object (specific), and
prototypic textures being typical for that kind of object and are used
many times for all objects of that kind (typical). A typical texture may
be replaced by a specific, if available. Textures may also be classified
as unknown. </xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="specific"/>
<xs:enumeration value="typical"/>
<xs:enumeration value="unknown"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
============== -->
<xs:simpleType name="Color">
<xs:annotation>
<xs:documentation>List of three values (red, green, blue), separated by
spaces. The values must be in the range between zero and one.
</xs:documentation>
```

```
</xs:annotation>
<xs:restriction base="core:doubleBetween0and1List">
<xs:length value="3"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
============== -->
<xs:simpleType name="ColorPlusOpacity">
<xs:annotation>
<xs:documentation>List of three or four values (red, green, blue,
opacity), separated by spaces. The values must be in the range between
zero and one. If no opacity is given, it is assumed as
1.0.</xs:documentation>
</xs:annotation>
<xs:restriction base="core:doubleBetween0and1List">
<xs:minLength value="3"/>
<xs:maxLength value="4"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

A.3. 桥梁模块

CityGML的*Bridge*模块在XML模式定义文件*bridge.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/bridge/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/bridge/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:core=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" targetNamespace=
"http://www.opengis.net/citygml/bridge/2.0" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractBridgeType" abstract="true">
<xs:annotation>
```

```
<xs:documentation>Type describing the thematic and geometric attributes
and the associations of bridges. It is an abstract type, only its
subclasses Bridge and BridgePart can be instantiated. An _AbstractBridge
may consist of BridgeParts, which are again _AbstractBridges by
inheritance. Thus an aggregation hierarchy between _AbstractBridges of
arbitrary depth may be specified. In such an hierarchy, top elements are
Bridges, while all other elements are BridgeParts. Each element of such a
hierarchy may have all attributes and geometries of _AbstractBridges. It
must, however, be assured that no inconsistencies
occur.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractSiteType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="yearOfConstruction" type="xs:gYear" minOccurs="0"/>
<xs:element name="yearOfDemolition" type="xs:gYear" minOccurs="0"/>
<xs:element name="isMovable" type="xs:boolean" default="false" minOccurs=
"0"/>
<xs:element name="lod1Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="outerBridgeConstruction" type=
"BridgeConstructionElementPropertyType" minOccurs="0" maxOccurs="
unbounded"/>
<xs:element name="outerBridgeInstallation" type=
"BridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="interiorBridgeInstallation" type=
"IntBridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="lod3Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
```

```
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="interiorBridgeRoom" type="
InteriorBridgeRoomPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="consistsOfBridgePart" type="BridgePartPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="address" type="core:AddressPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfAbstractBridge" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="_AbstractBridge" type="AbstractBridgeType" abstract=
"true" substitutionGroup="core:_Site"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAbstractBridge" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
=============== -->
<xs:complexType name="BridgeType">
<xs:complexContent>
<xs:extension base="AbstractBridgeType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfBridge" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="Bridge" type="BridgeType" substitutionGroup=
"_AbstractBridge"/>
```

```
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridge" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BridgePartType">
<xs:complexContent>
<xs:extension base="AbstractBridgeType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfBridgePart" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="BridgePart" type="BridgePartType" substitutionGroup=
"_AbstractBridge"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridgePart" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BridgePartPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its bridge
parts. The BridgePartPropertyType element must either carry a reference to
a BridgePart object or contain a BridgePart object inline, but neither
both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BridgePart"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="BridgeInstallationType">
<xs:annotation>
<xs:documentation>A BridgeInstallation is a part of a Bridge which has not
```

```
the significance of a BridgePart. In contrast to
BridgeConstructionElements, a BridgeInstallation is not essential from a
structural point of view. Thus, it may be removed without the bridge
collapsing. Examples are stairs, antennas, railways, etc. As subclass of
_CityObject, a BridgeInstallation inherits all attributes and relations,
in particular an id, names, external references, generic attributes and
generalization relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBridgeInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=========================================================================
============== -->
<xs:element name="BridgeInstallation" type="BridgeInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
=========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeInstallation" type=
"xs:anyType" abstract="true"/>
<!--
=========================================================================
============== -->
```

```
<xs:complexType name="BridgeInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its bridge
installations. The BridgeInstallationPropertyType element must either
carry a reference to a BridgeInstallation object or contain a
BridgeInstallation object inline, but neither both nor
none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BridgeInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="IntBridgeInstallationType">
<xs:annotation>
<xs:documentation>An IntBridgeInstallation is an interior part of a Bridge
which has a specific function or semantic meaning. Examples are interior
stairs, railings, radiators or pipes. As subclass of _CityObject, an
IntBridgeInstallation inherits all attributes and relations, in particular
an id, names, external references, generic attributes and generalization
relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfIntBridgeInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="IntBridgeInstallation" type="IntBridgeInstallationType"
```

```
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfIntBridgeInstallation"
type="xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="IntBridgeInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its
interior bridge installations. The IntBridgeInstallationPropertyType
element must either carry a reference to a IntBridgeInstallation object or
contain a IntBridgeInstallation object inline, but neither both nor
none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="IntBridgeInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:complexType name="BridgeConstructionElementType">
<xs:annotation>
<xs:documentation>A BridgeConstructionElement is a part of a Bridge which
has not the significance of a BridgePart. In contrast to
BridgeInstallation, a BridgeConstructionElement is essential from a
structural point of view. Examples are pylons, anchorages, etc. As
subclass of _CityObject, a BridgeInstallation inherits all attributes and
relations, in particular an id, names, external references, generic
attributes and generalization relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:annotation>
<xs:documentation> The name will be represented by gml:name (inherited
from _GML) The lodXMultiSurface must be used, if the geometry of a
building is just a collection of surfaces bounding a solid, but not a
topologically clean solid boundary necessary for GML3 solid boundaries.
</xs:documentation>
</xs:annotation>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
```

```
"unbounded"/>
<xs:element name="lod1Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBridgeConstructionElement"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="BridgeConstructionElement" type=
"BridgeConstructionElementType" substitutionGroup="core:_CityObject"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeConstructionElement"
type="xs:anyType"/>
<!--
========================================================================
=============== -->
<xs:complexType name="BridgeConstructionElementPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its bridge
```

```
construction elements. The BridgeConstructionElementPropertyType element
must either carry a reference to a BridgeConstructionElement object or
contain a BridgeConstructionElement object inline, but neither both nor
none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BridgeConstructionElement"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
<xs:annotation>
<xs:documentation>A BoundarySurface is a thematic object which classifies
surfaces bounding an _AbstractBridge, BridgeInstallation,
IntBuildingInstallation, BridgeConstructionElement, and BridgeRoom. The
geometry of a BoundarySurface is given by MultiSurfaces. As it is a
subclass of _CityObject, it inherits all atributes and relations, in
particular the external references, the generic attributes, and the
generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="opening" type="OpeningPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBoundarySurface" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
```

```
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="RoofSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfRoofSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="WallSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="WallSurface" type="WallSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
```

```
=============== -->
<xs:complexType name="GroundSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfGroundSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
=============== -->
<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
=============== -->
<xs:complexType name="ClosureSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfClosureSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
=============== -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
=============== -->
<xs:complexType name="OuterFloorSurfaceType">
<xs:complexContent>
```

```
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterFloorSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="OuterCeilingSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterCeilingSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="FloorSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfFloorSurface" minOccurs="0"
```

```
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="InteriorWallSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfInteriorWallSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="CeilingSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfCeilingSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
```

```
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="BoundarySurfacePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its
bounding thematic surfaces (walls, roofs, ..). The
BoundarySurfacePropertyType element must either carry a reference to a
_BoundarySurface object or contain a _BoundarySurface object inline, but
neither both nor none. There is no differentiation between interior
surfaces bounding rooms and outer ones bounding bridges (one reason is,
that ClosureSurface belongs to both types). It has to be made sure by
additional integrity constraints that, e.g. an _AbstractBridge is not
related to CeilingSurfaces or a room not to
RoofSurfaces.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_BoundarySurface"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="OpeningPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _BondarySurface to its
openings (doors, windows). The OpeningPropertyType element must either
carry a reference to an _Opening object or contain an _Opening object
inline, but neither both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_Opening"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
```

```
================================================================================
============== -->
<xs:complexType name="AbstractOpeningType" abstract="true">
<xs:annotation>
<xs:documentation> Type for openings (doors, windows) in boundary
surfaces. Used in LoD3 and LoD4 only. As subclass of _CityObject, an
_Opening inherits all attributes and relations, in particular an id,
names, external references, generic attributes and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfOpening" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="WindowType">
<xs:annotation>
<xs:documentation> Type for windows in boundary surfaces. Used in LoD3 and
LoD4 only . As subclass of _CityObject, a window inherits all attributes
and relations, in particular an id,names, external references, generic
attributes and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
```

```
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWindow" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="DoorType">
<xs:annotation>
<xs:documentation> Type for doors in boundary surfaces. Used in LoD3 and
LoD4 only . As subclass of _CityObject, a Door inherits all attributes and
relations, in particular an id, names, external references, generic
attributes and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
<xs:sequence>
<xs:element name="address" type="core:AddressPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfDoor" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
<!--
========================================================================
============== -->
```

```
<xs:complexType name="BridgeRoomType">
<xs:annotation>
<xs:documentation>A BridgeRoom is a thematic object for modelling the
closed parts inside a Bridge. It has to be closed, if necessary by using
closure surfaces. The geometry may be either a solid, or a MultiSurface if
the boundary is not topologically clean. The BridgeRoom connectivity may
be derived by detecting shared thematic openings or closure surfaces: two
rooms are connected if both use the same opening object or the same
closure surface. The thematic surfaces bounding a BridgeRoom are
referenced by the boundedBy property. As subclass of _CityObject,a
BridgeRoom inherits all attributes and relations, in particular an id,
names, external references, generic attributes and generalization
relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="interiorFurniture" type="InteriorFurniturePropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="bridgeRoomInstallation" type=
"IntBridgeInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBridgeRoom" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="BridgeRoom" type="BridgeRoomType" substitutionGroup=
"core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeRoom" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
```

```
============== -->
<xs:complexType name="BridgeFurnitureType">
<xs:annotation>
<xs:documentation>Type for bridge furnitures. As subclass of _CityObject,
a BridgeFurniture inherits all attributes and relations, in particular an
id, names, external references, generic attributes and generalization
relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfBridgeFurniture" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="BridgeFurniture" type="BridgeFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBridgeFurniture" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="InteriorBridgeRoomPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBridge to its rooms.
The InteriorBridgeRoomPropertyType element must either carry a reference
to an BridgeRoom object or contain an BridgeRoom object inline, but
neither both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BridgeRoom"/>
```

```
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="InteriorFurniturePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a BridgeRoom to its interior
bridge furniture. The InteriorBridgeFurniturePropertyType element must
either carry a reference to an BridgeFurniture object or contain an
BridgeFurniture object inline, but neither both nor
none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BridgeFurniture"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
</xs:schema>
```

A.4. 建筑模块

CityGML的*Building*模块在XML模式定义文件*building.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/building/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/building/2.0" xmlns:core
="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/building/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractBuildingType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the thematic and geometric attributes
```

```
and the associations of buildings. It is an abstract type, only its
subclasses Building and BuildingPart can be instantiated. An
_AbstractBuilding may consist of BuildingParts, which are again
_AbstractBuildings by inheritance. Thus an aggregation hierarchy between
_AbstractBuildings of arbitrary depth may be specified. In such an
hierarchy, top elements are Buildings, while all other elements are
BuildingParts. Each element of such a hierarchy may have all attributes
and geometries of _AbstractBuildings. It must, however, be assured than no
inconsistencies occur (for example, if the geometry of a Building does not
correspond to the geometries of its parts, or if the roof type of a
Building is saddle roof, while its parts have an hip roof). As subclass of
_CityObject, an _AbstractBuilding inherits all attributes and relations,
in particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractSiteType">
<xs:sequence>
<xs:annotation>
<xs:documentation> The name will be represented by gml:name (inherited
from _GML). list order for storeyHeightsAboveground: first floor, second
floor,... list order for storeyHeightsBelowground: first floor below
ground, second floor below ground,... The lodXMultiSurface must be used,
if the geometry of a building is just a collection of surfaces bounding a
solid, but not a topologically clean solid boundary necessary for GML3
solid boundaries. </xs:documentation>
</xs:annotation>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="yearOfConstruction" type="xs:gYear" minOccurs="0"/>
<xs:element name="yearOfDemolition" type="xs:gYear" minOccurs="0"/>
<xs:element name="roofType" type="gml:CodeType" minOccurs="0"/>
<xs:element name="measuredHeight" type="gml:LengthType" minOccurs="0"/>
<xs:element name="storeysAboveGround" type="xs:nonNegativeInteger"
minOccurs="0"/>
<xs:element name="storeysBelowGround" type="xs:nonNegativeInteger"
minOccurs="0"/>
<xs:element name="storeyHeightsAboveGround" type=
"gml:MeasureOrNullListType" minOccurs="0"/>
<xs:element name="storeyHeightsBelowGround" type=
"gml:MeasureOrNullListType" minOccurs="0"/>
<xs:element name="lod0FootPrint" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod0RoofEdge" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1Solid" type="gml:SolidPropertyType" minOccurs="0"/>
```

```
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="outerBuildingInstallation" type=
"BuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="interiorBuildingInstallation" type=
"IntBuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded
"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="lod3Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="interiorRoom" type="InteriorRoomPropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="consistsOfBuildingPart" type="BuildingPartPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="address" type="core:AddressPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfAbstractBuilding" minOccurs
="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="_AbstractBuilding" type="AbstractBuildingType" abstract
="true" substitutionGroup="core:_Site"/>
```

```
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfAbstractBuilding" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BuildingType">
<xs:complexContent>
<xs:extension base="AbstractBuildingType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfBuilding" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="Building" type="BuildingType" substitutionGroup=
"_AbstractBuilding"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBuilding" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BuildingPartType">
<xs:complexContent>
<xs:extension base="AbstractBuildingType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfBuildingPart" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="BuildingPart" type="BuildingPartType" substitutionGroup
="_AbstractBuilding"/>
<!--
================================================================================
=============== -->
```

```
<xs:element name="_GenericApplicationPropertyOfBuildingPart" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="BuildingPartPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBuilding to its
building parts. The BuildingPartPropertyType element must either carry a
reference to a BuildingPart object or contain a BuildingPart object
inline, but neither both nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BuildingPart"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="BuildingInstallationType">
<xs:annotation>
<xs:documentation>A BuildingInstallation is a part of a Building which has
not the significance of a BuildingPart. Examples are stairs, antennas,
balconies or small roofs. As subclass of _CityObject, a
BuildingInstallation inherits all attributes and relations, in particular
an id, names, external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
```

```
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBuildingInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="BuildingInstallation" type="BuildingInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBuildingInstallation" type
="xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="BuildingInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBuilding to its
building installations. The BuildingInstallationPropertyType element must
either carry a reference to a BuildingInstallation object or contain a
BuildingInstallation object inline, but neither both nor none.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BuildingInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:complexType name="IntBuildingInstallationType">
<xs:annotation>
<xs:documentation>An IntBuildingInstallation is an interior part of a
Building which has a specific function or semantical meaning. Examples are
interior stairs, railings, radiators or pipes. As subclass of _CityObject,
a nIntBuildingInstallation inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
```

```
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfIntBuildingInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="IntBuildingInstallation" type=
"IntBuildingInstallationType" substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfIntBuildingInstallation"
type="xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="IntBuildingInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBuilding to its
interior building installations. The IntBuildingInstallationPropertyType
element must either carry a reference to a IntBuildingInstallation object
or contain a IntBuildingInstallation object inline, but neither both nor
none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="IntBuildingInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
<xs:annotation>
<xs:documentation>A BoundarySurface is a thematic object which classifies
```

```
surfaces bounding an _AbstractBuilding, Room, BuildingInstallation, and
IntBuildingInstallation. The geometry of a BoundarySurface is given by
MultiSurfaces. As it is a subclass of _CityObject, it inherits all
atributes and relations, in particular the external references, and the
generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="opening" type="OpeningPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBoundarySurface" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
=============== -->
<xs:complexType name="RoofSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfRoofSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
```

```
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="WallSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="WallSurface" type="WallSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="GroundSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfGroundSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
```

```
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="ClosureSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfClosureSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="FloorSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfFloorSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
```

```
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="OuterFloorSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterFloorSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="InteriorWallSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfInteriorWallSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
```

```
=============== -->
<xs:complexType name="CeilingSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfCeilingSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="OuterCeilingSurfaceType">
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterCeilingSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BoundarySurfacePropertyType">
<xs:annotation>
```

```
<xs:documentation>Denotes the relation of an _AbstractBuilding to its
bounding thematic surfaces (walls, roofs, ..). The
BoundarySurfacePropertyType element must either carry a reference to a
_BoundarySurface object or contain a _BoundarySurface object inline, but
neither both nor none. There is no differentiation between interior
surfaces bounding rooms and outer ones bounding buildings (one reason is,
that ClosureSurface belongs to both types). It has to be made sure by
additional integrity constraints that, e.g. an _AbstractBuilding is not
related to CeilingSurfaces or a room not to RoofSurfaces.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_BoundarySurface"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="OpeningPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _BondarySurface to its
openings (doors, windows). The OpeningPropertyType element must either
carry a reference to an _Opening object or contain an _Opening object
inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_Opening"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="AbstractOpeningType" abstract="true">
<xs:annotation>
<xs:documentation> Type for openings (doors, windows) in boundary
surfaces. Used in LOD3 and LOD4 only. As subclass of _CityObject, an
_Opening inherits all attributes and relations, in particular an id,
names, external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
```

```
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfOpening" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="WindowType">
<xs:annotation>
<xs:documentation> Type for windows in boundary surfaces. Used in LOD3 and
LOD4 only . As subclass of _CityObject, a window inherits all attributes
and relations, in particular an id, names, external references, and
generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWindow" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
```

```
<!--
=======================================================================
=============== -->
<xs:complexType name="DoorType">
<xs:annotation>
<xs:documentation> Type for doors in boundary surfaces. Used in LOD3 and
LOD4 only . As subclass of _CityObject, a Door inherits all attributes and
relations, in particular an id, names, external references, and
generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
<xs:sequence>
<xs:element name="address" type="core:AddressPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfDoor" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="RoomType">
<xs:annotation>
<xs:documentation>A Room is a thematic object for modelling the closed
parts inside a building. It has to be closed, if necessary by using
closure surfaces. The geometry may be either a solid, or a MultiSurface if
the boundary is not topologically clean. The room connectivity may be
derived by detecting shared thematic openings or closure surfaces: two
rooms are connected if both use the same opening object or the same
closure surface. The thematic surfaces bounding a room are referenced by
the boundedBy property. As subclass of _CityObject, a Room inherits all
attributes and relations, in particular an id, names, external references,
and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
```

```
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="interiorFurniture" type="InteriorFurniturePropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="roomInstallation" type=
"IntBuildingInstallationPropertyType" minOccurs="0" maxOccurs="unbounded
"/>
<xs:element ref="_GenericApplicationPropertyOfRoom" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="Room" type="RoomType" substitutionGroup=
"core:_CityObject"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfRoom" type="xs:anyType"
abstract="true"/>
<!--
========================================================================
=============== -->
<xs:complexType name="BuildingFurnitureType">
<xs:annotation>
<xs:documentation>Type for building furnitures. As subclass of
_CityObject, a BuildingFurniture inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
```

```
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfBuildingFurniture"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="BuildingFurniture" type="BuildingFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBuildingFurniture" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="InteriorRoomPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractBuilding to its
rooms. The InteriorRoomPropertyType element must either carry a reference
to a Room object or contain a Room object inline, but neither both nor
none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="Room"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:complexType name="InteriorFurniturePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a Room to its interior
furnitures (movable). The InteriorFurniturePropertyType element must
either carry a reference to a BuildingFurniture object or contain a
BuildingFurniture object inline, but neither both nor none.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="BuildingFurniture"/>
```

```
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
</xs:schema>
```

A.5. 城市家具模块

CityGML的*CityFurniture*模块在XML模式定义文件*cityFurniture.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/cityfurniture/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/cityfurniture/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/cityfurniture/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="CityFurnitureType">
<xs:annotation>
<xs:documentation>Type describing city furnitures, like traffic lights,
benches, ... As subclass of _CityObject, a CityFurniture inherits all
attributes and relations, in particular an id, names, external references,
and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod1Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
```

```
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfCityFurniture" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="CityFurniture" type="CityFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityFurniture" type=
"xs:anyType" abstract="true"/>
</xs:schema>
```

A.6. 城市对象组模块

CityGML的*CityObjectGroup*模块在XML模式定义文件*cityObjectGroup.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/cityobjectgroup/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/cityobjectgroup/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
```

```
targetNamespace="http://www.opengis.net/citygml/cityobjectgroup/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<!--
================================================================================
=============== -->
<xs:complexType name="CityObjectGroupType">
<xs:annotation>
<xs:documentation> A group may be used to aggregate arbitrary CityObjects
according to some user-defined criteria. Examples for groups are the
buildings in a specific region, the result of a query, or objects put
together for visualization purposes. Each group has a name (inherited from
AbstractGMLType), functions (e.g., building group), a class and zero or
more usages. A geometry may optionally be attached to a group, if the
geometry of the whole group differs from the geometry of the parts. Each
member of a group may be qualified by a role name, reflecting the role
each CityObject plays in the context of the group. As subclass of
_CityObject, a CityObjectGroup inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. As CityObjectGroup itself is a CityObject, it may also be
contained by another group. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="groupMember" type="CityObjectGroupMemberType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="parent" type="CityObjectGroupParentType" minOccurs="0"/>
<xs:element name="geometry" type="gml:GeometryPropertyType" minOccurs="0
"/>
<xs:element ref="_GenericApplicationPropertyOfCityObjectGroup" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
```

```
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
============== -->
<xs:element name="CityObjectGroup" type="CityObjectGroupType"
substitutionGroup="core:_CityObject"/>
<!--
=======================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfCityObjectGroup" type=
"xs:anyType" abstract="true"/>
<!--
=======================================================================
============== -->
<xs:complexType name="CityObjectGroupMemberType">
<xs:annotation>
<xs:documentation>Denotes the relation of a CityObjectGroup to its
members, which are _CityObjects. The CityObjectGroupMemberType element
must either carry a reference to a _CityObject object or contain a
_CityObject object inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="core:_CityObject"/>
</xs:sequence>
<xs:attribute name="role" type="xs:string"/>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=======================================================================
============== -->
<xs:complexType name="CityObjectGroupParentType">
<xs:annotation>
<xs:documentation>Denotes the relation of a CityObjectGroup to its parent,
which is a CityObject. The CityObjectGroupParentType element must either
carry a reference to a _CityObject object or contain a _CityObject object
inline, but neither both nor none. The parent association allows for
modelling of a generic hierarchical grouping concept. Named aggregations
of components (CityObjects) can be added to specific CityObjects
considered as the parent object. The parent association links to the
aggregate, while the parts are given by the group members.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="core:_CityObject"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

```
</xs:schema>
```

A.7. 通用模块

CityGML的*Generics*模块在XML模式定义文件*generics.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/generics/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/generics/2.0" xmlns:core
="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/generics/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="GenericCityObjectType">
<xs:annotation>
<xs:documentation>Generic (user defined) city objects may be used to model
features which are not covered explicitly by the CityGML schema. Generic
objects must be used with care; they shall only be used if there is no
appropiate thematic class available in the overall CityGML schema.
Oherwise, problems concerning semantic interoperability may arise. As
subclass of _CityObject, a generic city object inherits all attributes and
relations, in particular an id, names, external references, and
generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod0Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod1Geometry" type="gml:GeometryPropertyType" minOccurs
```

```
="0"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod0TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod0ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="GenericCityObject" type="GenericCityObjectType"
substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:complexType name="AbstractGenericAttributeType" abstract="true">
<xs:annotation>
<xs:documentation> Generic (user defined) attributes may be used to
represent attributes which are not covered explicitly by the CityGML
schema. Generic attributes must be used with care; they shall only be used
if there is no appropiate attribute available in the overall CityGML
schema. Oherwise, problems concerning semantic interoperability may arise.
A generic attribute has a name and a value, which has further subclasses
(IntAttrribute, StringAttribute, ...). </xs:documentation>
</xs:annotation>
```

```
<xs:sequence/>
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_genericAttribute" type="AbstractGenericAttributeType"
abstract="true" substitutionGroup=
"core:_GenericApplicationPropertyOfCityObject"/>
<!--
================================================================================
============== -->
<xs:complexType name="StringAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element name="value" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="stringAttribute" type="StringAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
============== -->
<xs:complexType name="IntAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element name="value" type="xs:integer"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="intAttribute" type="IntAttributeType" substitutionGroup
="_genericAttribute"/>
```

```
<!--
================================================================================
============== -->
<xs:complexType name="DoubleAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element name="value" type="xs:double"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="doubleAttribute" type="DoubleAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
============== -->
<xs:complexType name="DateAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element name="value" type="xs:date"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="dateAttribute" type="DateAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
============== -->
<xs:complexType name="UriAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
```

```
<xs:sequence>
<xs:element name="value" type="xs:anyURI"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="uriAttribute" type="UriAttributeType" substitutionGroup
="_genericAttribute"/>
<!--
================================================================================
============== -->
<xs:complexType name="MeasureAttributeType">
<xs:annotation>
<xs:documentation/>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element name="value" type="gml:MeasureType"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="measureAttribute" type="MeasureAttributeType"
substitutionGroup="_genericAttribute"/>
<!--
================================================================================
============== -->
<xs:complexType name="GenericAttributeSetType">
<xs:annotation>
<xs:documentation>Set of generic attributes with an optional codeSpace. If
the codeSpace attribute is present, then its   value should identify an
authority for the set, such as the organisation or community who defined
its content. The generic attribute set may contain arbitrary generic
attributes.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractGenericAttributeType">
<xs:sequence>
<xs:element ref="_genericAttribute" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="codeSpace" type="xs:anyURI" use="optional"/>
</xs:extension>
```

```
</xs:complexContent>
</xs:complexType>
<!--
======================================================================
============== -->
<xs:element name="genericAttributeSet" type="GenericAttributeSetType"
substitutionGroup="_genericAttribute"/>
</xs:schema>
```

A.8. 土地利用模块

CityGML的*LandUse*模块在XML模式定义文件*landUse.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/landuse/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/landuse/2.0" xmlns:core=
"http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/landuse/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="LandUseType">
<xs:annotation>
<xs:documentation>Type describing the class for Land Use in all LOD.
LandUse objects describe areas of the earth's surface dedicated to a
specific land use. The geometry must consist of 3-D surfaces. As subclass
of _CityObject, a LandUse inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
```

```
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod0MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfLandUse" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="LandUse" type="LandUseType" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfLandUse" type="xs:anyType"
abstract="true"/>
</xs:schema>
```

A.9. 地形模块

CityGML的*Relief*模块在XML模式定义文件*relief.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/relief/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/relief/2.0" xmlns:core=
"http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/relief/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
```

```
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="ReliefFeatureType">
<xs:annotation>
<xs:documentation>Type describing the features of the Digital Terrain
Model. As subclass of _CityObject, a ReliefFeature inherits all attributes
and relations, in particular an id, names, external references, and
generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod" type="core:integerBetween0and4"/>
<xs:element name="reliefComponent" type="ReliefComponentPropertyType"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfReliefFeature" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="ReliefFeature" type="ReliefFeatureType"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfReliefFeature" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="AbstractReliefComponentType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the components of a relief feature -
either a TIN, a Grid, mass points or break lines. As subclass of
_CityObject, a ReliefComponent inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
```

```
<xs:sequence>
<xs:element name="lod" type="core:integerBetween0and4"/>
<xs:element name="extent" type="gml:PolygonPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfReliefComponent" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_ReliefComponent" type="AbstractReliefComponentType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfReliefComponent" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="ReliefComponentPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a ReliefFeature to its
components. The ReliefComponentPropertyType element must either carry a
reference to a _ReliefComponent object or contain a _ReliefComponent
object inline, but neither both nor
none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_ReliefComponent"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="TINReliefType">
<xs:annotation>
<xs:documentation>Type describing the TIN component of a relief feature.
As subclass of _CityObject, a TINRelief inherits all attributes and
relations, in particular an id, names, external references, and
generalization relations.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractReliefComponentType">
<xs:sequence>
<xs:element name="tin" type="tinPropertyType"/>
```

```
<xs:element ref="_GenericApplicationPropertyOfTinRelief" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="TINRelief" type="TINReliefType" substitutionGroup=
"_ReliefComponent"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTinRelief" type="
xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="RasterReliefType">
<xs:annotation>
<xs:documentation>Type describing the raster component of a relief
feature. As subclass of _CityObject, a RasterRelief inherits all
attributes and relations, in particular an id, names, external references,
and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractReliefComponentType">
<xs:sequence>
<xs:element name="grid" type="gridPropertyType"/>
<xs:element ref="_GenericApplicationPropertyOfRasterRelief" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="RasterRelief" type="RasterReliefType" substitutionGroup
="_ReliefComponent"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRasterRelief" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
```

```
=============== -->
<xs:complexType name="MassPointReliefType">
<xs:annotation>
<xs:documentation>Type describing the mass point component of a relief
feature. As subclass of _CityObject, a MassPoint Relief inherits all
attributes and relations, in particular an id, names, external references,
and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractReliefComponentType">
<xs:sequence>
<xs:element name="reliefPoints" type="gml:MultiPointPropertyType"/>
<xs:element ref="_GenericApplicationPropertyOfMassPointRelief" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="MassPointRelief" type="MassPointReliefType"
substitutionGroup="_ReliefComponent"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfMassPointRelief" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="BreaklineReliefType">
<xs:annotation>
<xs:documentation>Type describing the break line Component of a relief
feature. A break line relief consists of break lines or
ridgeOrValleyLines. As subclass of _CityObject, a BreaklineRelief inherits
all attributes and relations, in particular an id, names, external
references, and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractReliefComponentType">
<xs:sequence>
<xs:element name="ridgeOrValleyLines" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="breaklines" type="gml:MultiCurvePropertyType" minOccurs
="0"/>
<xs:element ref="_GenericApplicationPropertyOfBreaklineRelief" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
```

```
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="BreaklineRelief" type="BreaklineReliefType"
substitutionGroup="_ReliefComponent"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfBreaklineRelief" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="tinPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a TINRelief to its components.
The tinPropertyType element must either carry a reference to a
gml:TriangulatedSurface object or contain a gml:TriangulatedSurface object
inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="gml:TriangulatedSurface"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:complexType name="gridPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a RasterReliefType to its
components. The gridPropertyType element must either carry a reference to
a gml:RectifiedGridCoverage object or contain a gml:RectifiedGridCoverage
object inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="gml:RectifiedGridCoverage"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="Elevation" type="gml:LengthType" substitutionGroup=
"gml:_Object"/>
</xs:schema>
```

A.10. 交通模块

CityGML的*Transportation*模块在XML模式定义文件*transportation.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/transportation/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/transportation/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/transportation/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractTransportationObjectType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the abstract superclass for
transportation objects. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTransportationObject"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="_TransportationObject" type=
"AbstractTransportationObjectType" abstract="true" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTransportationObject" type
="xs:anyType" abstract="true"/>
```

```
<!--
================================================================================
============== -->
<xs:complexType name="TransportationComplexType">
<xs:annotation>
<xs:documentation>Type describing transportation complexes, which are
aggregated features, e.g. roads, which consist of parts (traffic areas,
e.g. pedestrian path, and auxiliary traffic areas). As subclass of
_CityObject, a TransportationComplex inherits all attributes and
relations, in particular an id, names, external references, and
generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTransportationObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="trafficArea" type="TrafficAreaPropertyType" minOccurs="
0" maxOccurs="unbounded"/>
<xs:element name="auxiliaryTrafficArea" type=
"AuxiliaryTrafficAreaPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="lod0Network" type="gml:GeometricComplexPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfTransportationComplex"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="TransportationComplex" type="TransportationComplexType"
substitutionGroup="_TransportationObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTransportationComplex"
```

```
type="xs:anyType" abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="TrafficAreaType">
<xs:annotation>
<xs:documentation>Type describing the class for traffic Areas. Traffic
areas are the surfaces where traffic actually takes place. As subclass of
_CityObject, a TrafficArea inherits all attributes and relations, in
particular an id, names,external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTransportationObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="surfaceMaterial" type="gml:CodeType" minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfTrafficArea" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:element name="TrafficArea" type="TrafficAreaType" substitutionGroup=
"_TransportationObject"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfTrafficArea" type=
"xs:anyType" abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="AuxiliaryTrafficAreaType">
<xs:annotation>
<xs:documentation>Type describing the class for auxiliary traffic Areas.
```

```
These are the surfaces where no traffic actually takes place, but which
belong to a transportation object. Examples are kerbstones, road markings
and grass stripes. As subclass of _CityObject, an AuxiliaryTrafficArea
inherits all attributes and relations, in particular an id, names,
external references, and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTransportationObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="surfaceMaterial" type="gml:CodeType" minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfAuxiliaryTrafficArea"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="AuxiliaryTrafficArea" type="AuxiliaryTrafficAreaType"
substitutionGroup="_TransportationObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfAuxiliaryTrafficArea" type
="xs:anyType" abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="TrafficAreaPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of TransportationComplex to its
parts, which are traffic areas. The TrafficAreaPropertyType element must
either carry a reference to a TrafficArea object or contain a TrafficArea
object inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="TrafficArea"/>
```

```
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:complexType name="AuxiliaryTrafficAreaPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of TransportationComplex to its
parts, which are auxiliary traffic areas. The TrafficAreaPropertyType
element must either carry a reference to a TrafficArea object or contain a
TrafficArea object inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="AuxiliaryTrafficArea"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:complexType name="TrackType">
<xs:annotation>
<xs:documentation>Type describing the class for tracks. A track is a small
path mainly used by pedestrians. As subclass of _CityObject, a Track
inherits all attributes and relations, in particular an id, names,
external references, and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="TransportationComplexType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTrack" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="Track" type="TrackType" substitutionGroup=
"TransportationComplex"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfTrack" type="xs:anyType"
abstract="true"/>
<!--
=============================================================================
```

```
============== -->
<xs:complexType name="RoadType">
<xs:annotation>
<xs:documentation>Type describing the class for roads. As subclass of
_CityObject, a Road inherits all attributes and relations, in particular
an id, names, external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="TransportationComplexType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfRoad" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="Road" type="RoadType" substitutionGroup=
"TransportationComplex"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRoad" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="RailwayType">
<xs:annotation>
<xs:documentation>Type describing the class for railways. As subclass of
_CityObject, a Railway inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="TransportationComplexType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfRailway" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
```

```
<xs:element name="Railway" type="RailwayType" substitutionGroup=
"TransportationComplex"/>
<!--
============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfRailway" type="xs:anyType"
abstract="true"/>
<!--
============================================================================
============== -->
<xs:complexType name="SquareType">
<xs:annotation>
<xs:documentation>Type describing the class for squares. A square is an
open area commonly found in cities (like a plaza).As subclass of
_CityObject, a Square inherits all attributes and relations, in particular
an id, names, external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="TransportationComplexType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfSquare" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
============== -->
<xs:element name="Square" type="SquareType" substitutionGroup=
"TransportationComplex"/>
<!--
============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfSquare" type="xs:anyType"
abstract="true"/>
```

A.11. 隧道模块

CityGML的*Tunnel*模块在XML模式定义文件*tunnel.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/tunnel/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/tunnel/2.0" xmlns:core=
"http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
```

```
targetNamespace="http://www.opengis.net/citygml/tunnel/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractTunnelType">
<xs:annotation>
<xs:documentation>Abstract super class of the features Tunnel and
TunnelPart</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractSiteType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="yearOfConstruction" type="xs:gYear" minOccurs="0"/>
<xs:element name="yearOfDemolition" type="xs:gYear" minOccurs="0"/>
<xs:element name="lod1Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod2Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod2TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="outerTunnelInstallation" type=
"TunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="interiorTunnelInstallation" type=
"IntTunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="lod3Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
```

```
minOccurs="0"/>
<xs:element name="lod3MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod3TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod4TerrainIntersection" type=
"gml:MultiCurvePropertyType" minOccurs="0"/>
<xs:element name="interiorHollowSpace" type=
"InteriorHollowSpacePropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="consistsOfTunnelPart" type="TunnelPartPropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfAbstractTunnel" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
==========================================================================
================ -->
<xs:element name="_AbstractTunnel" type="AbstractTunnelType" abstract=
"true" substitutionGroup="core:_Site"/>
<!--
==========================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfAbstractTunnel" type=
"xs:anyType" abstract="true"/>
<!--
==========================================================================
================ -->
<xs:complexType name="TunnelType">
<xs:annotation>
<xs:documentation>Horizontal or sloping underground or partly underground,
enclosed way of some length (ISO 6707-1)</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTunnelType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTunnel" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!--
================================================================================
================ -->
<xs:element name="Tunnel" type="TunnelType" substitutionGroup=
"_AbstractTunnel"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnel" type="xs:anyType"
abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="TunnelPartType">
<xs:annotation>
<xs:documentation>A Tunnel composed of structural segments differing in
important geometrical or semantical properties can be separated into one
Tunnel and additional TunnelParts.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractTunnelType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfTunnelPart" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="TunnelPart" type="TunnelPartType" substitutionGroup=
"_AbstractTunnel"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnelPart" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="TunnelPartPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _AbstractTunnel to its parts.
The TunnelPartPropertyType element must either carry a reference to a
TunnelPart object or contain a TunnelPart object inline, but neither both
nor none.</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
```

```
<xs:element ref="TunnelPart"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:complexType name="TunnelInstallationType">
<xs:annotation>
<xs:documentation>Immovable structural component of a Tunnel which has not
the significance of a TunnelPart. As subclass of
_CityObject, a TunnelInstallation inherits all attributes and relations,
in particular an id, names, external references, generic attributes and
generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfTunnelInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="TunnelInstallation" type="TunnelInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
```

```
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnelInstallation" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="TunnelInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a Tunnel to its external
installations. The TunnelInstallationPropertyType element must either
carry a reference to a TunnelInstallation object or contain a
TunnelInstallation object inline, but neither
both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="TunnelInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="IntTunnelInstallationType">
<xs:annotation>
<xs:documentation>Immovable interior structural component of a Tunnel or a
HollowSpace. Examples are interior stairs, railings, radiators or pipes.
As subclass of _CityObject, an IntTunnelInstallation inherits all
attributes and relations,in particular an id, names, external references,
and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfIntTunnelInstallation"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
```

```
</xs:complexContent>
</xs:complexType>
<!--
======================================================================
=============== -->
<xs:element name="IntTunnelInstallation" type="IntTunnelInstallationType"
substitutionGroup="core:_CityObject"/>
<!--
======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfIntTunnelInstallation"
type="xs:anyType" abstract="true"/>
<!--
======================================================================
=============== -->
<xs:complexType name="IntTunnelInstallationPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a Tunnel to its internal
installations. The InteriorTunnelInstallationPropertyType element must
either carry a reference to a IntTunnelInstallation object or contain a
IntTunnelInstallation object inline,but neither both nor none.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="IntTunnelInstallation"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
======================================================================
=============== -->
<xs:complexType name="BoundarySurfacePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an tunnel or hollow space to its
bounding thematic surfaces (walls, roofs, ..). There is no differentiation
between interior surfaces bounding hollow spaces and outer ones bounding
tunnels (one reason is, that ClosureSurface belongs to both types). It has
to be made sure by additional integrity constraints that, e.g. a tunnel is
not related to CeilingSurfaces or a room not to RoofSurfaces.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_BoundarySurface"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
======================================================================
=============== -->
```

```
<xs:complexType name="AbstractBoundarySurfaceType" abstract="true">
<xs:annotation>
<xs:documentation>Abstract super class of the features RoofSurface,
WallSurface, GroundSurface, ClosureSurface, FloorSurface,
OuterFloorSurface, CeilingSurface and
OuterCeilingSurface</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="opening" type="OpeningPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfBoundarySurface" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
=============== -->
<xs:element name="_BoundarySurface" type="AbstractBoundarySurfaceType"
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
========================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfBoundarySurface" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
=============== -->
<xs:complexType name="RoofSurfaceType">
<xs:annotation>
<xs:documentation>Construction that separates the interior part of the
Tunnel from the ambient medium ( water, air, ..) from
above.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfRoofSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
```

```
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
======================================================================
================ -->
<xs:element name="RoofSurface" type="RoofSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
======================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfRoofSurface" type=
"xs:anyType" abstract="true"/>
<!--
======================================================================
================ -->
<xs:complexType name="WallSurfaceType">
<xs:annotation>
<xs:documentation>Mainly vertical construction that separates the interior
part of the Tunnel from the ambient medium (rock, earth, water, air,
..)</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWallSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
======================================================================
================ -->
<xs:element name="WallSurface" type="WallSurfaceType" substitutionGroup=
"_BoundarySurface"/>
<!--
======================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
======================================================================
================ -->
<xs:complexType name="GroundSurfaceType">
<xs:annotation>
<xs:documentation>Horizontal construction that separates the interior part
of the Tunnel from the ambient medium (rock, earth, water, air, ..) from
below on the lowest level. </xs:documentation>
```

```
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfGroundSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="GroundSurface" type="GroundSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfGroundSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="ClosureSurfaceType">
<xs:annotation>
<xs:documentation>Virtual surface which can be used to define the volume
of geometric objects being not totally bounded by real surfaces (e. g. the
entrance of an open Tunnel).</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfClosureSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="ClosureSurface" type="ClosureSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfClosureSurface" type=
"xs:anyType" abstract="true"/>
<!--
```

```
================================================================================
================ -->
<xs:complexType name="FloorSurfaceType">
<xs:annotation>
<xs:documentation>Mostly horizontal construction that bounds a HollowSpace
from below. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfFloorSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="FloorSurface" type="FloorSurfaceType" substitutionGroup
="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfFloorSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="OuterFloorSurfaceType">
<xs:annotation>
<xs:documentation>Horizontal construction that separates the interior part
of the Tunnel from the ambient medium (rock, earth, water, air, ..) from
below.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterFloorSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="OuterFloorSurface" type="OuterFloorSurfaceType"
substitutionGroup="_BoundarySurface"/>
```

```
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfOuterFloorSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="InteriorWallSurfaceType">
<xs:annotation>
<xs:documentation>Mostly vertical construction that bounds a
HollowSpace.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfInteriorWallSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
================ -->
<xs:element name="InteriorWallSurface" type="InteriorWallSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
================================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfInteriorWallSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
================ -->
<xs:complexType name="CeilingSurfaceType">
<xs:annotation>
<xs:documentation>Mostly horizontal construction that bounds a HollowSpace
from above.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfCeilingSurface" minOccurs="
0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!--
=======================================================================
=============== -->
<xs:element name="CeilingSurface" type="CeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="OuterCeilingSurfaceType">
<xs:annotation>
<xs:documentation>Mainly horizontal construction that separates the
interior part of the Tunnel from the ambient medium (rock, earth, ..) from
above.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfOuterCeilingSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:element name="OuterCeilingSurface" type="OuterCeilingSurfaceType"
substitutionGroup="_BoundarySurface"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfOuterCeilingSurface" type=
"xs:anyType" abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="HollowSpaceType">
<xs:annotation>
<xs:documentation>Area or volume within a Tunnel bounded actually or
theoretically </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
```

```
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="boundedBy" type="BoundarySurfacePropertyType" minOccurs
="0" maxOccurs="unbounded"/>
<xs:element name="interiorFurniture" type="InteriorFurniturePropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="hollowSpaceInstallation" type=
"IntTunnelInstallationPropertyType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfHollowSpace" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="HollowSpace" type="HollowSpaceType" substitutionGroup=
"core:_CityObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfHollowSpace" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="InteriorHollowSpacePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a Tunnel to its internal hollow
spaces. The InteriorHollowSpacePropertyType element must either carry a
reference to a HollowSpace object or contain a HollowSpace object inline,
but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="HollowSpace"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:complexType name="TunnelFurnitureType">
```

```
<xs:annotation>
<xs:documentation>Movable, functional objects, whether useful or
ornamental, usually found in a HollowSpace</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfTunnelFurniture" minOccurs=
"0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
================ -->
<xs:element name="TunnelFurniture" type="TunnelFurnitureType"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
================ -->
<xs:element name="_GenericApplicationPropertyOfTunnelFurniture" type=
"xs:anyType" abstract="true"/>
<!--
========================================================================
================ -->
<xs:complexType name="InteriorFurniturePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a hollow space to the furnitures
it contains. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="TunnelFurniture"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
========================================================================
================ -->
<xs:complexType name="AbstractOpeningType" abstract="true">
```

```
<xs:annotation>
<xs:documentation> Type for openings (doors, windows) in boundary
surfaces. Used in LOD3 and LOD4 only. As subclass of
_CityObject, an _Opening inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfOpening" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="_Opening" type="AbstractOpeningType" abstract="true"
substitutionGroup="core:_CityObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfOpening" type="xs:anyType"
abstract="true"/>
<!--
========================================================================
============== -->
<xs:complexType name="OpeningPropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of an _BondarySurface to its
openings (doors, windows). The OpeningPropertyType element must either
carry a reference to an _Opening object or contain an _Opening object
inline, but neither both nor none. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_Opening"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

```
<!--
============================================================================
=============== -->
<xs:complexType name="WindowType">
<xs:annotation>
<xs:documentation>Construction for closing an _Opening not intended for
access or regress.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWindow" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
=============== -->
<xs:element name="Window" type="WindowType" substitutionGroup="_Opening"/>
<!--
============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWindow" type="xs:anyType"
abstract="true"/>
<!--
============================================================================
=============== -->
<xs:complexType name="DoorType">
<xs:annotation>
<xs:documentation>Construction for closing an _Opening intended primarily
for access or regress or both. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractOpeningType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfDoor" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
=============== -->
<xs:element name="Door" type="DoorType" substitutionGroup="_Opening"/>
<!--
============================================================================
```

```
=============== -->
<xs:element name="_GenericApplicationPropertyOfDoor" type="xs:anyType"
abstract="true"/>
</xs:schema>
```

A.12. 植被模块

CityGML的*Vegetation*模块在XML模式定义文件*vegetation.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/vegetation/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/vegetation/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/vegetation/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractVegetationObjectType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the abstract superclass for vegetation
objects. A subclass is either a SolitaryVegetationObject or a PlantCover.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfVegetationObject" minOccurs
="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
=============== -->
<xs:element name="_VegetationObject" type="AbstractVegetationObjectType"
```

```
abstract="true" substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfVegetationObject" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="PlantCoverType">
<xs:annotation>
<xs:documentation>Type describing Plant Covers resp. Biotopes. As subclass
of _CityObject, a VegetationObject inherits all attributes and relations,
in particular an id, names, external references, and generalization
relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractVegetationObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="averageHeight" type="gml:LengthType" minOccurs="0"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1MultiSolid" type="gml:MultiSolidPropertyType"
minOccurs="0"/>
<xs:element name="lod2MultiSolid" type="gml:MultiSolidPropertyType"
minOccurs="0"/>
<xs:element name="lod3MultiSolid" type="gml:MultiSolidPropertyType"
minOccurs="0"/>
<xs:element name="lod4MultiSolid" type="gml:MultiSolidPropertyType"
minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfPlantCover" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
```

```
================================================================================
=============== -->
<xs:element name="PlantCover" type="PlantCoverType" substitutionGroup=
"_VegetationObject"/>
<!--
================================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfPlantCover" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
=============== -->
<xs:complexType name="SolitaryVegetationObjectType">
<xs:annotation>
<xs:documentation>Type describing solitary vegetation objects, e.g.,
trees. Its geometry is either defined explicitly by a  GML 3 geometry with
absolute coordinates, or in the case of multiple occurences of the same
vegetation object, implicitly by a reference to a shape definition and a
transformation. The shape definition may be given in an external file. As
subclass of _CityObject, a SolitaryVegetationObject inherits all
attributes and relations, in particular an id, names, external references,
and generalization relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractVegetationObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="species" type="gml:CodeType" minOccurs="0"/>
<xs:element name="height" type="gml:LengthType" minOccurs="0"/>
<xs:element name="trunkDiameter" type="gml:LengthType" minOccurs="0"/>
<xs:element name="crownDiameter" type="gml:LengthType" minOccurs="0"/>
<xs:element name="lod1Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod2Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod3Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod4Geometry" type="gml:GeometryPropertyType" minOccurs
="0"/>
<xs:element name="lod1ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod2ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element name="lod3ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
```

```
<xs:element name="lod4ImplicitRepresentation" type=
"core:ImplicitRepresentationPropertyType" minOccurs="0"/>
<xs:element ref="_GenericApplicationPropertyOfSolitaryVegetationObject"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="SolitaryVegetationObject" type=
"SolitaryVegetationObjectType" substitutionGroup="_VegetationObject"/>
<!--
========================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfSolitaryVegetationObject"
type="xs:anyType" abstract="true"/>
</xs:schema>
```

A.13. 水体模块

CityGML的*WaterBody*模块在XML模式定义文件*waterBody.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/waterbody/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/waterbody/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/waterbody/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="AbstractWaterObjectType" abstract="true">
<xs:annotation>
<xs:documentation>Type describing the abstract superclass for water
objects. As subclass of _CityObject, a _WaterObject inherits all
```

```
attributes and relations, in particular an id, names, external references,
and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWaterObject" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="_WaterObject" type="AbstractWaterObjectType" abstract=
"true" substitutionGroup="core:_CityObject"/>
<!--
=============================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterObject" type=
"xs:anyType" abstract="true"/>
<!--
=============================================================================
============== -->
<xs:complexType name="WaterBodyType">
<xs:annotation>
<xs:documentation>Type describing Water Bodies, e.g., lakes, rivers. As
subclass of _CityObject, a WaterBody inherits all attributes and
relations, in particular an id, names, external references, and
generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractWaterObjectType">
<xs:sequence>
<xs:element name="class" type="gml:CodeType" minOccurs="0"/>
<xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs=
"unbounded"/>
<xs:element name="lod0MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod0MultiSurface" type="gml:MultiSurfacePropertyType"
minOccurs="0"/>
<xs:element name="lod1MultiCurve" type="gml:MultiCurvePropertyType"
minOccurs="0"/>
<xs:element name="lod1MultiSurface" type="gml:MultiSurfacePropertyType"
```

```
minOccurs="0"/>
<xs:element name="lod1Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod2Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod3Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="lod4Solid" type="gml:SolidPropertyType" minOccurs="0"/>
<xs:element name="boundedBy" type="BoundedByWaterSurfacePropertyType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="_GenericApplicationPropertyOfWaterBody" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="WaterBody" type="WaterBodyType" substitutionGroup=
"_WaterObject"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterBody" type="
xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="BoundedByWaterSurfacePropertyType">
<xs:annotation>
<xs:documentation>Denotes the relation of a WaterBody to its boundary
surfaces, which are of type _WaterBoundarySurface. The
BoundedByWaterSurfacePropertyType element must either carry a reference to
a _WaterBoundarySurface object or contain a
_WaterBoundarySurface object inline, but neither both nor none.
</xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_WaterBoundarySurface"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:complexType name="AbstractWaterBoundarySurfaceType" abstract="true">
<xs:annotation>
<xs:documentation>A WaterBoundarySurface is a thematic object which
classifies surfaces bounding a water body.
</xs:documentation>
</xs:annotation>
```

```
<xs:complexContent>
<xs:extension base="core:AbstractCityObjectType">
<xs:sequence>
<xs:element name="lod2Surface" type="gml:SurfacePropertyType" minOccurs="
0"/>
<xs:element name="lod3Surface" type="gml:SurfacePropertyType" minOccurs="
0"/>
<xs:element name="lod4Surface" type="gml:SurfacePropertyType" minOccurs="
0"/>
<xs:element ref="_GenericApplicationPropertyOfWaterBoundarySurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=======================================================================
=============== -->
<xs:element name="_WaterBoundarySurface" type=
"AbstractWaterBoundarySurfaceType" abstract="true" substitutionGroup=
"core:_CityObject"/>
<!--
=======================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWaterBoundarySurface" type
="xs:anyType" abstract="true"/>
<!--
=======================================================================
=============== -->
<xs:complexType name="WaterSurfaceType">
<xs:annotation>
<xs:documentation>Type describing the surface of a water body, which
separates the water from the air. As subclass of
_CityObject, a WaterSurface inherits all attributes and relations, in
particular an id, names, external references, and generalization
relations. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractWaterBoundarySurfaceType">
<xs:sequence>
<xs:element name="waterLevel" type="gml:CodeType" minOccurs="0">
<xs:annotation>
<xs:documentation>Type for the specification of the level of a water
surface. The optional attribute waterLevel of a WaterSurface can be used
to describe the water level, for which the given 3D surface geometry was
acquired. This is especially important, when the water body is influenced
by the tide. The values of this type are defined in the XML file
WaterLevelType.xml, according to the dictionary concept of GML3.
</xs:documentation>
```

```
</xs:annotation>
</xs:element>
<xs:element ref="_GenericApplicationPropertyOfWaterSurface" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="WaterSurface" type="WaterSurfaceType" substitutionGroup
="_WaterBoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterSurface" type=
"xs:anyType" abstract="true"/>
<!--
================================================================================
============== -->
<xs:complexType name="WaterGroundSurfaceType">
<xs:annotation>
<xs:documentation>Type describing the ground surface of a water body, i.e.
the boundary to the digital terrain model. As subclass of _CityObject, a
WaterGroundSurface inherits all attributes and relations, in particular an
id, names, external references, and generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractWaterBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWaterGroundSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
================================================================================
============== -->
<xs:element name="WaterGroundSurface" type="WaterGroundSurfaceType"
substitutionGroup="_WaterBoundarySurface"/>
<!--
================================================================================
============== -->
<xs:element name="_GenericApplicationPropertyOfWaterGroundSurface" type=
"xs:anyType" abstract="true"/>
<!--
```

```
============================================================================
=============== -->
<xs:complexType name="WaterClosureSurfaceType">
<xs:annotation>
<xs:documentation>Type describing the closure surface between water bodys.
As subclass of _CityObject, a WaterClosureSurface inherits all attributes
and relations, in particular an id, names, external references, and
generalization relations.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractWaterBoundarySurfaceType">
<xs:sequence>
<xs:element ref="_GenericApplicationPropertyOfWaterClosureSurface"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
============================================================================
=============== -->
<xs:element name="WaterClosureSurface" type="WaterClosureSurfaceType"
substitutionGroup="_WaterBoundarySurface"/>
<!--
============================================================================
=============== -->
<xs:element name="_GenericApplicationPropertyOfWaterClosureSurface" type=
"xs:anyType" abstract="true"/>
</xs:schema>
```

A.14. 纹理表面模块[已弃用]

CityGML的*TexturedSurface*模块在XML模式定义文件*texturedSurface.xsd*中被定义。目标命名空间 http://www.opengis.net/citygml/texturedsurface/2.0 与该扩展模块关联。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.opengis.net/citygml/texturedsurface/2.0"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:xs=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/citygml/texturedsurface/2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version
="2.0.0">
<xs:annotation>
<xs:documentation> CityGML is an OGC Standard.
Copyright (c) 2012 Open Geospatial Consortium.
To obtain additional rights of use, visit
```

```
http://www.opengeospatial.org/legal/ .
</xs:documentation>
</xs:annotation>
<xs:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation=
"http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xs:complexType name="TexturedSurfaceType">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. The concept of positioning
textures on surfaces complies with the standard X3D. Because there has
been no appropriate
texturing concept in GML3, CityGML adds the class TexturedSurface to the
geometry model of GML 3. A texture is specified as a raster image
referenced by an URI, and can be an arbitrary resource, even in the
internet. Textures are positioned by   employing the concept of texture
coordinates, i.e. each texture coordinate matches with exactly one 3D
coordinate of the TexturedSurface. The use of texture coordinates allows
an exact positioning and trimming of the texture on the surface  geometry.
Each surface may be assigned one or more appearances, each refering to one
side of the surface.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:OrientableSurfaceType">
<xs:sequence>
<xs:element ref="appearance" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
=============================================================================
============== -->
<xs:element name="TexturedSurface" type="TexturedSurfaceType"
substitutionGroup="gml:OrientableSurface"/>
<!--
=============================================================================
============== -->
<xs:element name="appearance" type="AppearancePropertyType"/>
<!--
=============================================================================
============== -->
<xs:complexType name="AppearancePropertyType">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. A property that has an
```

```
_Appearance as its value domain, which can either be a Material
(Color,...) or a Texture. The
_Appearance Element can either be encapsulated in an element of this type
or an XLink reference to a remote _Appearance element (where remote
geometry elements are located in another document or elsewhere in the same
document). Either the reference or the contained element must be given,
but neither both nor none. The side of the surface the _Appearance
refers to is given by the orientation attribute, which refers to the
corresponding sign attribute of the orientable
surface: + means the side with positive orientation, and - the side with
negative orientation. </xs:documentation>
</xs:annotation>
<xs:sequence minOccurs="0">
<xs:element ref="_Appearance"/>
</xs:sequence>
<xs:attribute name="orientation" type="gml:SignType" default="+"/>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:complexType name="AbstractAppearanceType" abstract="true">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. This abstract type is the parent
type of MaterialType and SimpleTextureType. It is derived from
gml:AbstractGMLType, thus it inherits the attribute gml:id and may be
referenced by an appearanceProperty, although it is defined elsewhere in
another appearanceProperty. </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="gml:AbstractGMLType"/>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="_Appearance" type="AbstractAppearanceType" abstract=
"true" substitutionGroup="gml:_GML"/>
<!--
========================================================================
============== -->
<xs:complexType name="MaterialType">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. Adopted from X3D standard
(http://www.web3d.org/x3d/) </xs:documentation>
```

```
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractAppearanceType">
<xs:sequence>
<xs:element name="shininess" type="core:doubleBetween0and1" minOccurs="0
"/>
<xs:element name="transparency" type="core:doubleBetween0and1" minOccurs=
"0"/>
<xs:element name="ambientIntensity" type="core:doubleBetween0and1"
minOccurs="0"/>
<xs:element name="specularColor" type="Color" minOccurs="0"/>
<xs:element name="diffuseColor" type="Color" minOccurs="0"/>
<xs:element name="emissiveColor" type="Color" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="Material" type="MaterialType" substitutionGroup=
"_Appearance"/>
<!--
========================================================================
============== -->
<xs:complexType name="SimpleTextureType">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. Adopted from X3D standard
(http://www.web3d.org/x3d/). ToDo: repeat </xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="AbstractAppearanceType">
<xs:sequence>
<xs:element name="textureMap" type="xs:anyURI"/>
<xs:element name="textureCoordinates" type="gml:doubleList"/>
<xs:element name="textureType" type="TextureTypeType" minOccurs="0"/>
<xs:element name="repeat" type="xs:boolean" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--
========================================================================
============== -->
<xs:element name="SimpleTexture" type="SimpleTextureType"
substitutionGroup="_Appearance"/>
<!--
```

```
================================================================================
============== -->
<xs:simpleType name="TextureTypeType">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead.Textures can be qualified by the
attribute textureType. The textureType differentiates between textures,
which are specific for a certain object and are only used for that object
(specific), and prototypic textures being typical for that kind of object
and are used many times for all objects of that kind (typical). A typical
texture may be replaced by a specific, if available. Textures may also be
classified as unknown. </xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="specific"/>
<xs:enumeration value="typical"/>
<xs:enumeration value="unknown"/>
</xs:restriction>
</xs:simpleType>
<!--
================================================================================
============== -->
<xs:simpleType name="Color">
<xs:annotation>
<xs:appinfo>deprecated</xs:appinfo>
<xs:documentation>Deprecated since CityGML version 0.4.0. Use the concepts
of the CityGML Appearance module instead. List of three values (red,
green, blue), separated by spaces. The values must be in the range between
zero and one.</xs:documentation>
</xs:annotation>
<xs:restriction base="core:doubleBetween0and1List">
<xs:length value="3"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

A.15. 关于引用完整性的模式化规则

CityGML模式库提供了模式化（Schematron）规则，精确地描述了施加在CityGML属性元素上的引用完整性约束，表示了CityGML对象之间的关系。如果无另外说明，这些属性元素通常需借助GML 3.1.1的XLink概念对远程对象进行引用（其中远程对象位于另一个文档或同一文档的其他地方），或其内部包含一个对象，但两种方式不能同时使用，也不能采取其他方式。在本标准文档中，关于引用完整性的一致性要求，以单独条款的形式在，在涵盖CityGML模块的每个章节的结尾进行了说明（例如，参见CityGML *Building*模块的章节10.3.9）。

Schematron是一种辅助约束性语言。这些约束适用于CityGML实例文档（通常是XML元素）中基于XPath标准进行寻址的上下文节点。为检查其有效性，每一个节点中的以XPath表达式形式给定的规则均被评估。如果任何一条Schematron规则被违反，CityGML实例文档将被视为无效。请注意，对于CityGML 2.0，提供

的Schematron规则仅允许属性元素以引用或值的形式存在，但两种形式不能同时存在，也不能以其它形式存在。对属性元素的进一步限制和其它一致性要求都没有涵盖。在CityGML的未来版本中，可能会额外补充新的Schematron规则。

在CityGML中，用于表达引用完整性规则的Schematron模式是基于Schematron Assertion语言的1.5版本发展而来，GML 3.1.1也采用了该语言。该模式是作为CityGML模式包（schema package）中的一个单独的文件，并可通过名称*referentialIntegrity.sch*访问。它可用于根据附录B中提供的CityGML抽象测试套件检查CityGML实例文档中CityGML属性元素的一致性要求。然而，这需要一个XML验证器，以能够自动处理前述模式所提供的Schematron规则。否则，Schematron代码只能被视为所需的约束条件的正式描述性语言。

下面的*referentialIntegrity.sch*摘录阐述了*hrefOrContent*抽象基本规则。此规则描述了XPath表达式，以确保属性元素的目标对象是通过引用或值的形式给出的。它与GML 3.1.1的XSD模式文件*gmlBase.xsd*提供的相应规则相同。由于*hrefOrContent*规则是抽象的，它不适用于CityGML实例文档中任何具体的上下文节点，而是被模式中的具体规则所引用以确保一致性。

```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://www.ascc.net/xml/schematron">
<sch:title>Schematron constraints for CityGML 2.0</sch:title>
<sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
<sch:ns prefix="bldg" uri="http://www.opengis.net/citygml/building/2.0"/>
…
<sch:pattern name="Check either href or content not both">
<sch:rule abstract="true" id="hrefOrContent">
<sch:report test="@xlink:href and (*|text())">Property element may not
carry both a reference to an object and contain an object.</sch:report>
<sch:assert test="@xlink:href | (*|text())">Property element must either
carry a reference to an object or contain an object.</sch:assert>
</sch:rule>
</sch:pattern>
…
</sch:schema>
```

对于受引用完整性约束的每个CityGML属性元素，*referentialIntegrity.sch*包含基于*hrefOrContent*抽象规则的更多具体规则。例如，以下模式片段将*hrefOrContent*规则应用于CityGML的建筑模块中定义的属性元素*bldg:consistsOfBuildingPart*（参见第10.3章）。*bldg:consistsOfBuildingPart*元素表示*bldg:AbstractBuilding*与其建筑部件对象之间的关系，这些对象只能通过包含或引用的方式给出。

```
<sch:pattern name="hrefOrContent check on bldg:consistsOfBuildingPart">
<sch:rule context="bldg:consistsOfBuildingPart">
<sch:extends rule="hrefOrContent"/>
</sch:rule>
</sch:pattern>
…
```

Annex B: （规范性）CityGML实例文档的抽象测试套件

B.1. 强制性一致性要求的测试用例

B.1.1. 有效的CityGML实例文档

a)测试目的	对于实例文档所采用的CityGML专用文件部分的每个CityGML模块，依据它们的XML模式定义，验证CityGML实例文档的有效性。
b)测试方法	依据所有被采用的CityGML模块的XML模式定义，验证CityGML的XML实例文档的有效性。该过程可能使用合适的软件工具进行验证，或者通过人工依据每个被采用的CityGML模块所对应的XML模式说明书检查所有相关的定义。
c)参考	附录A
d)测试类型	基础测试

B.1.2. 有效的CityGML专用文件

嵌入在CityGML实例文档的CityGML专用文件定义

a)测试目的	依据章节7.2的规则和指南，验证CityGML实例文档所采用的专用文件是有效的CityGML专用文件。对于内嵌在ityGML实例文档中的CityGML专用文件定义（章节7.2提及的“第一种方法”），请验证CityGML实例文件是否表示所有的模式定义，及相应的CityGML模块的XML命名空间。其中这些CityGML模块被用作表示实例文档中的数据，并且是被采用的CityGML专用文件的一部分。
b)测试方法	审查实例文档，并检查它是否符合章节7.2（第一种方法）所描述的采用CityGML专用文件所需遵循的规则。
c)参考	附录A，章节7.2（第一种方法）
d)测试类型	基础测试

由独立XML模式定义文件提供的CityGML专用文件定义

a)测试目的	依据章节7.2所述的规则和指引，验证CityGML实例文档采用的专用文件是有效的CityGML专用文件。对于由单独的XML模式定义文件（在章节7.2中作为第二种方法引用）提供的CityGML专用文件定义，验证专用文件的XML模式定义文件本身是否有效，并导入用于表达实例文档数据的CityGML模块的所有模式定义。因此，它们也是所采用的CityGML专用文件的一部分。专用文件的XML模式定义的目标命名空间必须与导入的CityGML模块的命名空间不同，并且必须以之前未被使用且全局唯一的URI形式给出。专用文件的XML架构定义不得包含任何其他内容。
b)测试方法	验证CityGML专用文件的XML模式定义。该过程可以使用合适的软件工具进行验证，也可以手动从CityGML专用文件相应的XML模式规范中检查所有相关定义。检查实例文档并检查它是否满足章节7.2（第二种方法）中描述的使用CityGML专用文件的规则。
c)参考	附录A，章节7.2（第二种方法）
d)测试类型	基础测试

B.1.3. 与CityGML模块相关的一致性类

a)测试目的	根据每个CityGML模块的一致性类验证CityGML实例文档的有效性。这些模块是实例文档采用的CityGML专用文件的一部分，可以是CityGML扩展模块与CityGML核心模块的任意组合。
b)测试方法	遵循附录B.2中每个CityGML模块的一致性类所提供的测试用例。
c)参考	附录B.2
d)测试类型	基础测试

B.1.4. 空间几何对象

a)测试目的	验证CityGML实例文档中的所有空间几何对象是否符合地理标记语言（GML）版本3.1.1的XML模式定义，是否符合CityGML空间模型。
b)测试方法	检查实例文档，并基于GML版本3.1.1的XML模式定义，核验空间几何对象是否有效，及是否满足章节8中描述的CityGML空间模型的规则。
c)参考	附录B.2
d)测试类型	能力测试

B.1.5. 空间拓扑关系

a)测试目的	验证空间几何对象之间的所有空间拓扑关系是否使用GML 3.1.1版提供的XML概念XLinks来表示。
b)测试方法	检查实例文档，并基于GML 3.1.1版引入的*XLinks*概念核验空间几何对象之间的空间拓扑关系是否有效，及是否满足第8章中描述的CityGML空间模型的规则。
c)参考	OGC 文件编号 03-105r1，附录A，章节8
d)测试类型	能力测试

B.1.6. 地址对象

a)测试目的	验证在CityGML实例文档中表示地址信息的所有主题对象是否符合OASIS发布的可扩展地址语言 (xAL)的XML模式定义，以及是否符合CityGML中地址信息的表达规则。
b)测试方法	检查实例文档，并基于OASIS可扩展地址语言的XML模式定义，核验表示地址信息的主题对象是否有效，及是否满足章节10.1.4中描述的 CityGML地址信息的表达规则。
c)参考	OASIS 2003，附件A，章节10.1.4
d)测试类型	能力测试

B.2. 与CityGML模块相关的一致性类

B.2.1. CityGML核心模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*CityGML Core*模块的对象和属性编码规则，及是否遵守其所有一致性要求。此测试用例对于所有CityGML实例文档都是必须的。
b)测试方法	检查实例文档并检查它是否满足章节10.1和章节8.2中描述的*CityGML Core*模块的规则，尤其是章节10.1.6和章节8.3.3中所述的一致性要求。*CityGML Core*模块中定义的CityGML属性元素的引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束进行额外验证。
c)参考	章节10.1、章节10.1.6、章节8.2、章节8.3.3，附录A.15
d)测试类型	基础测试

有效的CityGML实例文档

a)测试目的	根据*CityGML Core*模块的XML模式定义，验证CityGML实例文档的有效性。此测试用例对于所有CityGML实例文档都是必须的。
b)测试方法	根据附录A.1中*CityGML Core*模块的XML模式定义验证CityGML XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*CityGML Core*模块中所有相关定义。
c)参考	附录A.1
d)测试类型	基础测试

B.2.2. 外观模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Appearance*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Appearance*模块中定义的元素的CityGML实例文档都是强制性的。*Appearance*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节9中描述的*Appearance*模块的规则，尤其是章节9.7中所述的一致性要求。
c)参考	章节9，章节9.7，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Appearance*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Appearance*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.2中*Appearance*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Appearance*模块中所有相关的定义。
c)参考	附录A.2
d)测试类型	能力测试

B.2.3. 桥梁模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Bridge*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Bridge*模块中定义的元素的CityGML实例文档都是强制性的。*Bridge*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.5中描述的*Bridge*模块的规则，尤其是章节10.5.8中所述的一致性要求。
c)参考	章节10.5，章节10.5.8，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Bridge*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Bridge*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.3中*Bridge*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Bridge*模块中所有相关的定义。
c)参考	附录A.3
d)测试类型	能力测试

B.2.4. 建筑模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Building*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Building*模块中定义的元素的 CityGML实例文档都是强制性的。*Building*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。

b)测试方法	检查实例文档，并检查它是否满足章节10.3中描述的*Building*模块的规则，尤其是章节10.3.9中所述的一致性要求。
c)参考	章节10.3，章节10.3.9，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Building*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Building*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.3中*Building*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Building*模块中所有相关的定义。
c)参考	附录A.4
d)测试类型	能力测试

B.2.5. 城市家具模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*CityFurniture*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*CityFurniture*模块中定义的元素的CityGML实例文档都是强制性的。*CityFurniture*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.9中描述的*CityFurniture*模块的规则，尤其是章节10.9.4中所述的一致性要求。
c)参考	章节10.9，章节10.9.4，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*CityFurniture*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*CityFurniture*模块中定义的元素的CityGML实例文档都是强制性的。

b)测试方法	根据附件A.5中*CityFurniture*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*CityFurniture*模块中所有相关的定义。
c)参考	附录A.5
d)测试类型	能力测试

B.2.6. 城市对象组模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*CityObjectGroup*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*CityObjectGroup*模块中定义的元素的CityGML实例文档都是强制性的。*CityObjectGroup*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.11中描述的*CityObjectGroup*模块的规则，尤其是章节10.11.2中所述的一致性要求。
c)参考	章节10.11，章节10.11.2，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*CityObjectGroup*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*CityObjectGroup*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.6中*CityObjectGroup*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*CityObjectGroup*模块中所有相关的定义。
c)参考	附录A.6
d)测试类型	能力测试

B.2.7. 通用模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Generics*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Generics*模块中定义的元素的CityGML实例文档都是强制性的。*Generics*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.12中描述的*Generics*模块的规则，尤其是章节10.12.2中所述的一致性要求。
c)参考	章节10.12，章节10.12.2，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Generics*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Generics*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.7中通用模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Generics*模块中所有相关的定义。
c)参考	附录A.7
d)测试类型	能力测试

B.2.8. 土地利用模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*LandUse*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*LandUse*模块中定义的元素的CityGML实例文档都是强制性的。*LandUse*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.10中描述的*LandUse*模块的规则，尤其是章节10.10.3中所述的一致性要求。
c)参考	章节10.10，章节10.10.3，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*LandUse*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*LandUse*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.8中*LandUse*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*LandUse*模块中所有相关的定义。
c)参考	附录A.8
d)测试类型	能力测试

B.2.9. 地形模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Relief*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Relief*模块中定义的元素的CityGML实例文档都是强制性的。*Relief*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.2中描述的*Relief*模块的规则，尤其是章节10.2.6中所述的一致性要求。
c)参考	章节10.2，章节10.2.6，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Relief*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Relief*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.9中*Relief*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Relief*模块中所有相关的定义。
c)参考	附录A.9
d)测试类型	能力测试

B.2.10. 交通模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Transportation*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Transportation*模块中定义的元素的CityGML实例文档都是强制性的。*Transportation*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.7中描述的*Transportation*模块的规则，尤其是章节10.7.5中所述的一致性要求。
c)参考	章节10.7，章节10.7.5，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Transportation*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Transportation*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.10中*Transportation*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Transportation*模块中所有相关的定义。
c)参考	附录A.10
d)测试类型	能力测试

B.2.11. 隧道模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Tunnel*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Tunnel*模块中定义的元素的CityGML实例文档都是强制性的。*Tunnel*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch提*供的约束，进行额外验证。

b)测试方法	检查实例文档，并检查它是否满足章节10.4中描述的*Tunnel*模块的规则，尤其是章节10.4.8中所述的一致性要求。
c)参考	章节10.4，章节10.4.8，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Tunnel*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Tunnel*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.3中*Tunnel*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Tunnel*模块中所有相关的定义。
c)参考	附录A.11
d)测试类型	能力测试

B.2.12. 植被模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*Vegetation*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*Vegetation*模块中定义的元素的CityGML实例文档都是强制性的。*Vegetation*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.8中描述的*Vegetation*模块的规则，尤其是章节10.8.6中所述的一致性要求。
c)参考	章节10.8，章节10.8.6，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*Vegetation*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*Vegetation*模块中定义的元素的CityGML实例文档都是强制性的。

b)测试方法	根据附件A.12中*Vegetation*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*Vegetation*模块中所有相关的定义。
c)参考	附录A.12
d)测试类型	能力测试

B.2.13. 水体模块

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*WaterBody*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*WaterBody*模块中定义的元素的CityGML实例文档都是强制性的。*WaterBody*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节10.6中描述的*WaterBody*模块的规则，尤其是章节10.6.4中所述的一致性要求。
c)参考	章节10.6，章节10.6.4，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*WaterBody*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*WaterBody*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.13中*WaterBody*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*WaterBody*模块中所有相关的定义。
c)参考	附录A.13
d)测试类型	能力测试

B.2.14. 带纹理的表面模块[已弃用]

强制性的一致性要求

a)测试目的	验证CityGML实例文档是否遵循*TexturedSurface*模块的对象和属性编码规则并遵守其所有一致性要求。该测试用例对于所有使用了*TexturedSurface*模块中定义的元素的CityGML实例文档都是强制性的。*TexturedSurface*模块中定义的CityGML属性元素引用完整性的一致性要求，可以根据附录A.15中规定的规则和指南，使用Schematron模式*referentialIntegrity.sch*提供的约束，进行额外验证。
b)测试方法	检查实例文档，并检查它是否满足章节9.8中描述的*TexturedSurface*模块的规则，尤其是章节9.8.2中所述的一致性要求。
c)参考	章节9.8，章节9.8.2，附录A.15
d)测试类型	能力测试

有效的CityGML实例文档

a)测试目的	根据*TexturedSurface*模块的XML模式定义验证CityGML实例文档的有效性。该测试用例对于所有使用了*TexturedSurface*模块中定义的元素的CityGML实例文档都是强制性的。
b)测试方法	根据附件A.14中*TexturedSurface*模块的XML模式定义验证CityGML的XML实例文档。该过程可能使用合适的软件工具进行验证，也可能是人工检查*TexturedSurface*模块中所有相关的定义。
c)参考	附录A.14
d)测试类型	能力测试

Annex C: （资料性附录）SIG 3D提议的代码清单

本附件举例说明了*gml:CodeType*类型的枚举属性的代码列表规范，并为选定的属性提供了建议。请注意，本附件是非规范性的，所提供的代码列表既不是强制性的，也不是完整的。代码列表的管理与CityGML模式和规范的管理是明确相互分离的（参见章节10.14）。因此，任何组织或信息社区都可以根据自己的信息需要，在CityGML规范之外自由定义代码列表，且每个代码列表都应有一个负责维护和更新内容的官方组织。

本附件中提供的代码列表由SIG 3D管理和维护，可从 http://www.sig3d.de/codelists/standard/ 访问。它们完全包含之前CityGML 1.0规范文档中给出的非规范代码列表（参见 OGC Doc. No. 08-007r1的附件C）。为了保持向后兼容性，之前的条目都没有被修改或改变含义。对于CityGML 2.0中新引入的一些枚举属性，已有额外的代码列表被添加进来，例如*Bridge*和*Tunnel*的模块便是如此。SIG 3D 代码列表将之后将继续补充，先前的1.0版代码列表仍可从位于 http://schemas.opengis.net/citygml/codelists/ 的OGC存储库中被访问。

SIG 3D代码列表以遵循*GML 3.1.1 Simple Dictionary Profile*的*simple dictionaries*形式被实现，并可在CityGML实例文档中被直接引用。为此，由*gml:CodeType*声明的*codeSpace*属性必须指向对应的枚举特征属性的代码列表。SIG 3D应用以下命名模式，以便在http模式中为每个代码列表生成一个全局唯一的URL：

http://www.sig3d.org/codelists/standard/
<moduleName>/<version>/<FeatureTypeName>_<propertyName>.xml

占位符*<moduleName>*必须被替换为定义要素和属性的CityGML模块的名称，模块名称必须以小写字母给出（例如，*building*或*cityfurniture*），*<version>*字段定义指定代码列表的CityGML版本。最后，*<FeatureTypeName>*表示挂接属性的要素类型名称，*<propertyName>*将被替换为属性名称本身。对于这两个占位符，指定要素的CityGML模式文档中所列出的XML元素名称，必须要被选定好。

例如，用于标识*_AbstractBuilding*要素（*Building*模块，参见章节10.3）枚举属性*roofType*的SIG 3D代码列表的*codeSpace*值是：

http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_roofType.xml

以下片段演示了如何从CityGML实例文档中引用*roofType*的SIG 3D代码列表。根据此代码列表（参见子条款C.1），属性值*1000*表示*flat roof*。

```
<bldg:Building>
<bldg:roofType codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_ro
ofType.xml">1000</bldg:roofType>
…
</bldg:Building>
```

可以从这些URL下载SIG 3D代码列表，以允许应用程序自动验证被编码的属性值。

建筑屋顶类型的整个代码列表在下文以范例的形式给出，而其他类型的代码列表则由于篇幅原因，在以下子条款中以表格形式进行描述。子条款根据主题模块进行排序。

屋顶类型的代码列表（*http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_roofType.xml*）

```
<gml:Dictionary xmlns:gml="http://www.opengis.net/gml" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/gml
http://schemas.opengis.net/gml/3.1.1/profiles/SimpleDictionary/1.0.0/gmlSi
mpleDictionaryProfile.xsd" gml:id="_AbstractBuilding_roofType">
<gml:name>_AbstractBuilding_roofType</gml:name>
<gml:dictionaryEntry>
<gml:Definition gml:id="id1">
<gml:description>flat roof</gml:description>
<gml:name>1000</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id2">
<gml:description>monopitch roof</gml:description>
<gml:name>1010</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id3">
<gml:description>skip pent roof</gml:description>
<gml:name>1020</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id4">
<gml:description>gabled roof</gml:description>
<gml:name>1030</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id5">
<gml:description>hipped roof</gml:description>
<gml:name>1040</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id6">
<gml:description>half-hipped roof</gml:description>
<gml:name>1050</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id7">
<gml:description>mansard roof</gml:description>
<gml:name>1060</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
```

```
<gml:dictionaryEntry>
<gml:Definition gml:id="id8">
<gml:description>pavilion roof</gml:description>
<gml:name>1070</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id9">
<gml:description>cone roof</gml:description>
<gml:name>1080</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id10">
<gml:description>copula roof</gml:description>
<gml:name>1090</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id11">
<gml:description>shed roof</gml:description>
<gml:name>1100</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id12">
<gml:description>arch roof</gml:description>
<gml:name>1110</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id13">
<gml:description>pyramidal broach roof</gml:description>
<gml:name>1120</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="id14">
<gml:description>combination of roof forms</gml:description>
<gml:name>1130</gml:name>
</gml:Definition>
</gml:dictionaryEntry>
</gml:Dictionary>
```

C.1. 建筑模块

***_AbstractBuilding*的*class*属性的代码列表**

http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_class.xml			
1000	habitation	1100	schools, education, research
1010	sanitation	1110	maintainence and waste management
1020	administration	1120	healthcare
1030	business, trade	1130	communicating
1040	catering	1140	security
1050	recreation	1150	storage
1060	sport	1160	industry
1070	culture	1170	traffic
1080	church institution	1180	function
1090	agriculture, forestry		

***_AbstractBuilding*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_function.xml http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_usage.xml			
1000	residential building	1840	rubbish bunker
1010	tenement	1850	building for rubbish incineration
1020	hostel	1860	building for rubbish disposal
1030	residential- and administration building	1870	building for agrarian and forestry
1040	residential- and office building	1880	barn
1050	residential- and business building	1890	stall
1060	residential- and plant building	1900	equestrian hall
1070	agrarian- and forestry building	1910	alpine cabin
1080	residential- and commercial building	1920	hunting lodge
1090	forester’ s lodge	1930	arboretum
1100	holiday house	1940	glass house
1110	summer house	1950	moveable glass house

1120	office building	1960	public building
1130	credit institution	1970	administration building
1140	insurance	1980	parliament
1150	business building	1990	guildhall
1160	department store	2000	post office
1170	shopping centre	2010	customs office
1180	kiosk	2020	court
1190	pharmacy	2030	embassy or consulate
1200	pavilion	2040	district administration
1210	hotel	2050	district government
1220	youth hostel	2060	tax office
1230	campsite building	2070	building for education and research
1240	restaurant	2080	comprehensive school
1250	cantine	2090	vocational school
1260	recreational site	2100	college or university
1270	function room	2110	research establishment
1280	cinema	2120	building for cultural purposes
1290	bowling alley	2130	castle
1300	casino	2140	theatre or opera
1310	industrial building	2150	concert building
1320	factory	2160	museum
1330	workshop	2170	broadcasting building
1340	petrol / gas station	2180	activity building
1350	washing plant	2190	library
1360	cold store	2200	fort
1370	depot	2210	religious building
1380	building for research purposes	2220	church
1390	quarry	2230	synagogue
1400	salt works	2240	chapel
1410	miscellaneous industrial building	2250	community center

1420	mill	2260	place of worship
1430	windmill	2270	mosque
1440	water mill	2280	temple
1450	bucket elevator	2290	convent
1460	weather station	2300	building for health care
1470	traffic assets office	2310	hospital
1480	street maintenance	2320	healing centre or care home
1490	waiting hall	2330	health centre or outpatients clinic
1500	signal control box	2340	building for social purposes
1510	engine shed	2350	youth centre
1520	signal box or stop signal	2360	seniors centre
1530	plant building for air traffic	2370	homeless shelter
1540	hangar	2380	kindergarten or nursery
1550	plant building for shipping	2390	asylum seekers home
1560	shipyard	2400	police station
1570	dock	2410	fire station
1580	plant building for canal lock	2420	barracks
1590	boathouse	2430	bunker
1600	plant building for cablecar	2440	penitentiary or prison
1610	multi-storey car park	2450	cemetery building
1620	parking level	2460	funeral parlor
1630	garage	2470	crematorium
1640	vehicle hall	2480	train station
1650	underground garage	2490	airport building
1660	building for supply	2500	building for underground station
1670	waterworks	2510	building for tramway
1680	pump station	2520	building for bus station

1690	water basin	2530	shipping terminal
1700	electric power station	2540	building for recuperation purposes
1710	transformer station	2550	building for sport purposes
1720	converter	2560	sports hall
1730	reactor	2570	building for sports field
1740	turbine house	2580	swimming baths
1750	boiler house	2590	indoor swimming pool
1760	building for telecommunications	2600	sanatorium
1770	gas works	2610	zoo building
1780	heat plant	2620	green house
1790	pumping station	2630	botanical show house
1800	building for disposal	2640	bothy
1810	building for effluent disposal	2650	tourist information centre
1820	building for filter plant	2700	others
1830	toilet		

***BuildingFurniture*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/BuildingFurniture_class.xml			
1000	habitation	1100	schools, education, research
1010	sanitation	1110	maintenance, waste management
1020	administration	1120	healthcare
1030	business, trade	1130	communicating
1040	catering	1140	security
1050	recreation	1150	storage
1060	sport	1160	industry
1070	culture	1170	traffic
1080	church institution	1180	function
1090	agriculture, forestry		

***BuildingFurniture*的*function*和*usage*属性的代码列表**

http://www.sig3d.org/codelists/standard/building/2.0/BuildingFurniture_function.xml http://www.sig3d.org/codelists/standard/building/2.0/BuildingFurniture_usage.xml			
1000	cupboard	2010	sink, hand-basin
1010	wardrobe	2020	water tap
1020	cabinet	2030	toilet bowl
1030	sideboard	2040	bathtub
1040	locker	2050	shower
1050	tool cabinet	2060	bidet
1100	shelf	2100	animal park
1110	rack	2110	aquarium
1120	coat stand	2120	cage
1200	table	2130	birdcage
1210	dining table	2200	religious equipment
1220	coffee table	2300	shop fittings
1230	desk	2310	sales counter
1240	bedside cabinet	2320	glass cabinet
1250	baby changing table	2330	changing cubicle
1260	bar	2340	refrigerated counter
1270	pool table	2350	cash desk or till or counter
1280	snooker table	2360	box-office
1290	roulette table	2400	machines
1270	work bench	2410	ticket machine
1300	chair	2420	cigarette machine
1310	bench	2430	cash machine or ATM
1320	office chair	2440	vending machine
1330	sofa	2450	gambling machine
1340	rocking chair	2500	technical furniture
1350	bar stool	2510	heating installation
1360	armchair	2520	tank
1400	bed	2521	oil tank
1410	crib	2522	water tank
1420	bunk bed	2523	gas tank
1430	cradle	2524	fuel tank

1440	cot	2525	milk tank
1450	stretcher	2526	steel tank
1500	lighting	2530	fire protection appliance
1510	standard lamp	2531	fire extinguishing system
1520	ceiling light	2532	fire alarm
1530	spotlight	2533	fire extinguisher
1600	electric appliances	2540	switch board
1610	television set	2550	lifting platform
1620	video recorder	2560	compressed air system
1630	stereo unit	2570	loud-speaker
1700	kitchen appliances	2580	microphone
1710	cooker	2600	sports equipment
1720	oven	2610	goal posts
1730	refrigerator	2620	basketball basket
1740	coffee machine	2630	volleyball net
1750	toaster	2640	gymnastic apparatus
1760	kettle	2650	diving platform
1770	microwave	2660	swimming pool
1780	dish washer	2700	sales promotion furniture
1800	laundry equipment	2710	display panel
1810	washing machine	2720	billboard
1820	ironing machine	2730	display cabinet
1830	rotary iron (mangle)	2800	functional furniture
1840	laundry tumble drier	2805	ashtray
1850	spin drier	2810	lectern
1900	technical office equipment	2815	stage
1910	copy machine	2820	blackboard
1920	scanner	2825	screen
1930	plotter	2830	mapstand
1940	printer	2835	rubbish bin
1950	screen	2840	sauna
1960	computer	2845	carpet

1970	overhead projector	2850	wall clock
1980	video projector	2855	curtain
2000	sanitation equipment	2860	mirror

***BuildingInstallation*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/BuildingInstallation_class.xml			
1000	outer characteristics	1040	communicating
1010	inner characteristics	1050	security
1020	waste management	1060	others
1030	maintenance		

***BuildingInstallation*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/BuildingInstallation_function.xml http://www.sig3d.org/codelists/standard/building/2.0/BuildingInstallation_usage.xml			
1000	balcony	1040	tower (part of a building)
1010	winter garden	1050	column
1020	arcade	1060	stairs
1030	chimney (part of a building)	1070	others

***IntBuildingInstallation*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/IntBuildingInstallation_class.xml			
1000	Heating, Ventilation, Climate	6000	Statics
2000	Safety	7000	Entertainmant
3000	Illumination	8000	Miscellaneous
4000	Communication	9999	Unknown
5000	Supply and Disposal		

***IntBuildingInstallation*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/IntBuildingInstallation_function.xml http://www.sig3d.org/codelists/standard/building/2.0/IntBuildingInstallation_usage.xml			
1010	Radiator	3020	Light switch
1020	Oven	5030	Power point
1030	Fireside	5020	Cable
1040	Ventilator	7010	Rafter

1050	Air Conditioning	7020	Column
5010	Pipe	8010	Railing
3010	Lamp	8020	Stair

***_AbstractBuilding*的*roofType*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_roofType.xml			
1000	flat roof	1070	pavilion roof
1010	monopitch roof	1080	cone roof
1020	dual pent roof	1090	copula roof
1030	gabled roof	1100	sawtooth roof
1040	hipped roof	1110	arch roof
1050	half-hipped roof	1120	pyramidal broach roof
1060	mansard roof	1130	combination of roof forms

***Room*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/Room_class.xml			
1000	habitation	1080	accommodation, waste management
1010	administration	1090	healthcare
1020	business, trade	1100	communicating
1030	catering	1110	security
1040	recreation	1120	store
1050	church institution	1130	industry
1060	agriculture, forestry	1140	traffic
1070	schools, education, research	1150	function

***Room*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/building/2.0/Room_function.xml http://www.sig3d.org/codelists/standard/building/2.0/Room_usage.xml			
1000	living room	2170	showers
1010	bedroom	2200	tribune
1020	kitchen	2210	seating / standing capacity
1030	hall	2220	cash point

1040	bath, washroom	2230	vivarium
1050	toilet	2240	enclosure
1060	stairs	2250	aquarium
1070	home office	2260	terrarium
1080	utility room	2270	aviary
1090	dining room	2280	menagerie
1100	common room	2290	stables
1110	party room	2300	greenhouse
1120	nursery	2310	food silo
1130	store room	2320	hayloft
1140	canteen, common kitchen	2330	motor pool
1150	storeroom	2340	barn
1160	balcony, gallery	2350	riding hall
1170	terrace	2360	horse box
1180	drying room	2370	hunting lodge
1190	heatingroom	2400	waste container
1200	fuel depot	2410	motor pool
1210	hobby room	2420	washing-bay
1220	stable, hovel	2430	installations room
1300	cash office	2440	monitoring room
1310	ticket office	2450	heating system
1320	conference room	2460	public utility use
1330	reception	2470	pump room
1340	sales room	2480	effluent treatment
1350	store room	2490	treatment installation
1360	delivery	2500	recycling installation
1370	lounge, common room	2600	chancel
1380	escalator	2610	sacristy
1390	guest toilet	2620	bell tower
1400	strong room	2630	baptism room
1500	office	2640	confessional
1510	entrance hall	2650	benches

1520	elevator	2660	pulpit
1530	canteen	2670	lobby
1540	tea kitchen / Coffee kitchen	2680	parish
1550	archive	2690	chapel
1560	citizen office	2700	police station
1570	conference hall	2710	headquarters
1580	copier room / blueprint room	2720	prison cell
1590	information	2730	motor pool hall
1600	computer room	2740	fire brigade, emergency vehicle
1610	printer / plotter room	2750	relaxation room
1700	reception	2760	tool / pipe store
1710	guest room	2770	emergency call center
1720	bar	2780	arms depot
1730	breakfast room	2790	ammunition dump
1740	dining room	2800	vehicle hall
1750	celebration room	2810	panic room
1760	pub	2900	satellite receiver
1770	beer garden	2910	communication room
1780	restaurant	3000	industrial building
1790	cool store	3010	production building
1800	bowling alley, shoot alley	3020	factory building
1810	lounge	3030	workshop
1820	canteen kitchen	3040	storage depot
1900	stage	3050	cold storage
1910	auditorium	3060	store
1920	VIP box	3100	station concourse
1930	projection room	3110	track
1940	dressing room	3120	ticket office
1950	cabin	3130	waiting hall
1960	showroom	3140	engine shed
1970	equipment or props	3150	signal box

1980	make-up room	3160	departure terminal
1990	recording studio	3170	check-out counter
2000	sound studio	3180	check-in counter
2010	music archive	3190	check
2020	administration	3200	baggage carousel
2030	ticket office	3210	security check
2040	library	3300	classroom
2050	media room	3310	staff room
2060	dressing room	3320	break or recess hall
2070	sport room	3330	laboratory
2080	equipment room	3340	utility room
2090	platform	3350	media room
2100	swimming-pool	3360	science laboratory
2110	slide	3370	sports hall
2120	relaxation room	3380	school library
2130	sauna	3390	office
2140	fitness room	3400	lecture theatre
2150	solarium	3410	refectory
2160	catering	3420	function room

C.2. 隧道模块

***_AbstractTunnel*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/tunnel/2.0/_AbstractTunnel_class.xml			
1000	traffic	1030	others
1010	supply		
1020	historical		

***_AbstractTunnel*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/tunnel/2.0/_AbstractTunnel_function.xml http://www.sig3d.org/codelists/standard/tunnel/2.0/_AbstractTunnel_usage.xml			
1000	railway tunnel	1020	canal tunnel
1010	roadway tunnel	1030	pedestrian tunnel

C.3. 桥梁模块

*_AbstractBridge*的*class*属性的代码列表			
http://www.sig3d.org/codelists/standard/bridge/2.0/_AbstractBridge_class.xml			
1000	arced bridge	1040	truss bridge
1010	cable-stayed bridge	1050	pontoon bridge
1020	deck bridge	1060	suspension bridge
1030	cable-stayed overpass		

*_AbstractBridge*的*function*和*usage*属性的代码列表			
http://www.sig3d.org/codelists/standard/bridge/2.0/_AbstractBridge_function.xml http://www.sig3d.org/codelists/standard/bridge/2.0/_AbstractBridge_usage.xml			
1000	railway bridge	1040	canal bridge
1010	roadway bridge	1050	aqueduct
1030	cable link	1060	foot bridge

C.4. 城市家具模块

*CityFurniture*的*class*属性的代码列表			
http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_class.xml			
1000	traffic	1020	security
1010	communication	1030	others

*CityFurniture*的*function*和*usage*属性的代码列表			
http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_function.xml http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_usage.xml			
1000	communication fixture	1270	pole
1010	telephone box	1280	radio mast
1020	postbox	1290	aerial
1030	emergency call fixture	1300	radio telescope
1040	fire detector	1310	chimney
1050	police call post	1320	marker
1060	switching unit	1330	hydrant
1070	road sign	1340	upper corridor fire-hydrant

1080	traffic light	1350	lower floor panel fire-hydrant
1090	free-standing sign	1360	slidegate valve cap
1100	free-standing warning sign	1370	entrance shaft
1110	bus stop	1380	converter
1120	milestone	1390	stair
1130	rail level crossing	1400	outside staircase
1140	gate	1410	escalator
1150	streetlamp, latern or candelabra	1420	ramp
1160	column	1430	patio
1170	lamp post	1440	fence
1180	flagpole	1450	memorial/monument
1190	street sink box	1470	wayside shrine
1200	rubbish bin	1480	crossroads
1210	clock	1490	cross on the summit of a mountain
1220	directional spot light	1500	fountain
1230	floodlight mast	1510	block mark
1240	windmill	1520	boundary post
1250	solar cell	1530	bench
1260	water wheel	1540	others

C.5. 土地利用模块

***LandUse*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/landuse/2.0/LandUse_class.xml			
1000	Settlement Area	3000	Vegetation
1100	Undeveloped Area	4000	Water
2000	Traffic		

***LandUse*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/landuse/2.0/LandUse_function.xml http://www.sig3d.org/codelists/standard/landuse/2.0/LandUse_usage.xml			
1010	Residential	2050	Track

1020	Industry and Business	2060	Square
1030	Mixed use	3010	Grassland
1040	Special Function Area	3020	Agriculture
1050	Monument	3030	Forest
1060	Dump	3040	Grove
1070	Mining	3050	Heath
1110	Park	3060	Moor
1120	Cemetary	3070	Marsh
1130	Sports, leisure and recreation	3080	Untilled land
1140	Open pit, quarry	4010	River
2010	Road	4020	Standing Waterbody
2020	Railway	4030	Harbour
2030	Airfield	4040	Sea
2040	Shipping		

C.6. 矿物模块

***ImplicitGeometry*（*Core*模块）和*Texture*（*Appearance*模块）的*mimeType*属性的代码列表**			
http://www.sig3d.org/codelists/standard/core/2.0/ImplicitGeometry_mimeType.xml http://www.sig3d.org/codelists/standard/appearance/2.0/_Texture_mimeType.xml			
此表中给出的MIME类型由Internet号码分配机构（IANA）定义。请参阅 http://www.iana.org/。 通常，MIME格式由Internet工程任务组（IETF）进行标准化工作。请参阅 http://www.ietf.org/ 。与其他代码列表不同，MIME类型并非由数字表示，而是使用它们给定的标识符。此代码列表并不详尽。它仅包含一系列常用的MIME类型。			
model/vrml	VRML97	model/x3d+xml	X3D
application/x-3ds	3ds max	model/x3d+binary	X3D
application/dxf	AutoCad DXF	image/gif	*.gif images
application/x-autocad	AutoCad DXF	image/jpeg	*.jpeg, *.jpg images
application/x-dxf	AutoCad DXF	image/png	*.png images
application/acad	AutoCad DWG	image/tiff	*.tiff, *.tif images
application/x-shockwave-flash	Shockwave 3D	image/bmp	*.bmp images
model/x3d+vrml	X3D		

C.7. 植被模块

***SolitaryVegetationObject*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetationObject_class.xml			
1000	shrub	1060	coniferous tree
1010	low plants	1070	decidous tree
1020	medium high plants	1080	bushes
1030	high plants	1090	aquatic plants
1040	grasses	1100	climber
1050	ferns	9999	unknown

***SolitaryVegetationObject*的*function*和*usage*属性的代码列表**
http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetationObject_function.xml http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetationObject_usage.xml
代码列表与*SolitaryVegetationObject*的*class*属性相同。

***PlantCover*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/vegetation/2.0/PlantCover_class.xml			
1010	Lemnetea	1280	Arrhenatheretea
1020	Asplenietea rupestris	1290	Molinio-Juncetea
1030	Adiantetea	1300	Scheuchzerio-Caricetea fuscae azidophile
1040	Thlaspietea rotundifolii	1310	Festuco-Brometea
1050	Crithmo-Limonietea	1320	Elyno-Seslerietea
1060	Ammophietea	1330	Caricetea curvulae azidophile
1070	Cakiletea maritimae halophile	1340	Calluno-Ulicetea
1080	Secalinetea	1350	Oxycocco-Sphagnetea
1090	Chenopodietea	1360	Salicetea purpureae
1100	Onopordetea	1370	Betulo-Adenostyletea
1110	Epilobietea angustifolii	1380	Alnetea glutinosae
1120	Bidentetea tripartiti	1390	Erico-Pinetea
1130	Zoosteretea marinae halophile	1400	Vaccinio-Piceetea

1140	Ruppietea maritimae	1410	Quercetea robori-petraeae
1150	Potametea haftende	1420	Querco-Fagetea
1160	Litorelletea	1430	Crithmo-Staticetea
1170	Plantaginetea majoris	1440	Tuberarietea guttati
1180	Isoeto-Nanojuncetea	1450	Juncetea maritimae
1190	Montino-Cardaminetea	1460	Thero-Brachypodietea
1200	Corynephoretea	1470	Ononido-Rosmarinetea
1210	Asteretea tripolium	1480	Nerio-Tamaricetea
1220	Salicornietea	1490	Pegano-Salsoletea
1230	Juncetea maritimi	1500	Cisto-Lavanduletea
1240	Phragmitetea	1510	Quercetea ilicis
1250	Spartinetea	1520	Populetea albae
1260	Sedo-Scleranthetea	9999	unknown
1270	Salicetea herbaceae		

***PlantCover*的*function*和*usage*属性的代码列表**
http://www.sig3d.org/codelists/standard/vegetation/2.0/PlantCover_function.xml http://www.sig3d.org/codelists/standard/vegetation/2.0/PlantCover_usage.xml
代码列表与*PlantCover*的class__属性相同。

***SolitaryVegetationObject*的*species*属性的代码列表（节选）**			
http://www.sig3d.org/codelists/standard/vegetation/2.0/SolitaryVegetationObject_species.xml			
1640	Abies alba	1790	Acer circinatum
1650	Abies cephalonica	1800	Acer Davidii
1660	Abies concolor	1810	Acer ginnala Maxim
1670	Abies grandis	1820	Acer grosserii
1680	Abies homolepsis	1830	Acer monspessulanum
1690	Abies koreana	1840	Acer negundo
1700	Abies lasiocarpa	1850	Acer palmatum
1710	Abies nordmanniana	1860	Acer platanoides
1720	Abies pinsapo	1870	Acer platanoides 'Crimson King'
1730	Abies procera	1880	Acer pseudoplatanus
1740	Abies procera 'Glauca'	1890	Acer rubrum

1750	Abies veitchii	1900	Acer saccharinum
1760	Acer campéstre	1910	Acer saccharum Marsch
1770	Acer capillipes	1920	Acer tartaricum
1780	Acer cappadocicum	…	…

C.8. 交通模块

***AuxiliaryTrafficArea*的*function*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/AuxiliaryTrafficArea_function.xml			
1000	soft shoulder	1300	traffic island
1010	hard shoulder	1400	bank
1020	green area	1410	embankment, dike
1030	middle lane	1420	railroad embankment
1040	lay by	1430	noise protection
1100	parking bay	1440	noise protection wall
1200	ditch	1500	noise guard bar
1210	drainage	1600	towpath
1220	kerbstone	1700	others
1230	flower tub		

***TrafficArea*的*function*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/TrafficArea_function.xml			
1	driving_lane	20	crosswalk
2	footpath	21	barrier
3	cyclepath	22	stairs
4	combined foot-/cyclepath	23	escalator
5	square	24	filtering lane
6	car_park	25	airport_runway
7	parking_lay_by	26	airport_taxiway
8	rail	27	airport_apron
9	rail_road_combined	28	airport_heliport
10	drainage	29	airport_runway_marking
11	road marking	30	green spaces
12	road_marking_direction	31	recreation

13	road_marking_lane	32	bus_lay_by
14	road_marking_restricted	33	motorway
15	road_marking_crosswalk	34	motorway_entry
16	road_marking_stop	35	motorway_exit
17	road_marking_other	36	motorway_emergency lane
18	overhead wire (trolley)	37	private_area
19	train platform	9999	unknown

***TrafficArea*的*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/TrafficArea_usage.xml			
1	pedestrian	9	boat, ferry, ship
2	car	10	teleferic
3	truck	11	aeroplane
4	bus, taxi	12	helicopter
5	train	13	taxi
6	bicycle	14	horse
7	motorcycle	9999	unknown
8	tram, streetcar		

***TrafficArea*的*AuxiliaryTrafficArea*和*surfaceMaterial*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/TrafficArea_surfaceMaterial.xml			
1	asphalt	8	soil
2	concrete	9	sand
3	pavement	10	grass
4	cobblestone	11	wood
5	gravel	12	steel
6	rail_with_bed	13	marble
7	rail_without_bed	9999	unknown

***TransportationComplex*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/TransportationComplex_class.xml			
1000	private	1050	air traffic
1010	common	1060	rail traffic
1020	civil	1070	waterway

1030	military	1080	subway
1040	road traffic	1090	others

***TransportationComplex*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/transportation/2.0/TransportationComplex_function.xml http://www.sig3d.org/codelists/standard/transportation/2.0/TransportationComplex_usage.xml			
1000	road	1855	railway track
1010	freeway/motorway	1860	magnetic levitation train
1020	highway/national primary road	1900	railway station
1030	land road	1910	stop
1040	district road	1920	station
1050	municipal road	2000	power-wheel
1060	main through-road	2100	airport
1100	freeway interchange/ highway junction	2110	international airport
1110	junction	2120	regional airport
1200	road	2130	landing place
1210	driveway	2140	heliport
1220	footpath/footway	2150	landing place
1230	hiking trail	2160	gliding airfield
1240	bikeway/cycle-path	2170	taxiway
1250	bridleway/bridlepath	2180	apron
1260	main agricultural road	2190	runway
1270	agricultural road	2200	canal
1280	bikeway/footway	2300	harbor
1290	access road	2310	pleasure craft harbour
1300	dead-end road	2400	ferry
1400	lane	2410	car ferry
1410	lane, one direction	2420	train ferry
1420	lane, both direction	2430	ferry
1500	pedestrian zone	2500	landing stage
1600	place	2600	waterway I order
1610	parking area	2610	navigable river

1620	marketplace	2620	inland navigation waterway 0
1700	service area	2621	inland navigation waterway 0
1800	rail transport	2622	inland navigation waterway I
1805	rail	2623	inland navigation waterway II
1810	urban/city train	2624	inland navigation waterway III
1815	city railway	2625	inland navigation waterway IV
1820	tram	2626	inland navigation waterway V
1825	subway	2627	inland navigation waterway VI
1830	funicular/mountain railway	2628	inland navigation waterway VII
1835	mountain railway	2630	maritime navigation
1840	chairlift	2640	navigable lake
1845	ski-lift/ski tow lift	2700	others
1850	suspension railway		

C.9. 水体模块

***WaterBody*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/waterbody/2.0/WaterBody_class.xml			
1000	sea	1140	flooded land
1010	tidal waterbody	1150	artificial waterbody
1020	watercourse	1160	aqueduct
1030	river / stream	1170	canal
1040	ditch	1180	port basin
1050	spring / water hole	1190	reservior
1060	lake / pont	1200	excavation pont
1070	bayou	1210	moat
1080	body of standing water	1220	pool

1090	waterfall	1230	fountain
1100	rapids	1240	well
1110	swamp	1250	cistern
1120	sinkhole (karst)	1260	fish ladder
1130	ephemeral watercourse	9999	unknown

***WaterBody*的*function*属性的代码列表**			
http://www.sig3d.org/codelists/standard/waterbody/2.0/WaterBody_function.xml			
1000	nature-sanctuary	1090	public swimming
1010	protected waterbody	1100	public fountain
1020	reservoir	1110	private waterbody
1030	retention waterbody	1120	irrigation waterbody
1040	flood plain waterbody	1130	watering place
1050	waterway	1140	industrial waterbody
1060	habor waterbody	1150	waterbody for fire-fighting
1070	sluice waterbody	9999	unknown
1080	sewage system		

***WaterBody*的*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/waterbody/2.0/WaterBody_usage.xml			
1000	sanctuary	1110	industrial / craft water supply
1010	recreation / sports	1120	military use
1020	drinking water supply	1130	mining / excavation
1030	hydroelectric water supply	1140	irrigation water supply
1040	ocean shipping	1150	fishing water
1050	inland shipping	1160	fish farm
1060	sewer	1170	archaeological site
1070	port	1180	water protection area
1080	anchorage	1190	abandoned
1090	public use	9999	unknown
1100	private use		

***WaterBody*的*waterLevel*属性的代码列表**			
http://www.sig3d.org/codelists/standard/waterbody/2.0/WaterSurface_waterLevel.xml			
1000	MSL - Mean Sea Level	1090	Hundred Year Flood
1010	LAT - Lowest Astronomical Tide	1100	highest known water level
1020	National Water Level	1110	critical low-water level
1030	Mean High Tide (related to National Waterlevel)	1120	lowest known water level
1040	Extreme High Tide (related to National Waterlevel)	1130	Established Line of Navigability
1050	Mean Low Tide (related to National Waterlevel)	1140	Minimum Limit of Navigability
1060	Extreme Low Tide (related to National Waterlevel)	1150	Maximum Limit of Navigability
1070	Mean Water Level (watercourse)	9999	unknown
1080	critical high-water level		

C.10. 城市对象组模块

***CityObjectGroup*的*class*属性的代码列表**			
http://www.sig3d.org/codelists/standard/cityobjectgroup/2.0/CityObjectGroup_class.xml			
1000	building separation	2000	assembly

***CityObjectGroup*的*function*和*usage*属性的代码列表**			
http://www.sig3d.org/codelists/standard/cityobjectgroup/2.0/CityObjectGroup_function.xml http://www.sig3d.org/codelists/standard/cityobjectgroup/2.0/CityObjectGroup_usage.xml			
1000	lod1Storey	1020	lod3Storey
1010	lod2Storey	1030	lod4Storey

Annex D: （资料性附录）采用的GML3几何类概述

Abstract GML classes referenced in CityGML	GML subclass actually used in CityGML
_Geometry	
_Solid	Solid (boundary is restricted to OrientableSurfaces, TexturedSurfaces, Polygons or CompositeSurfaces)
	CompositeSolid
_Surface	Polygon (with holes, modelled by _Rings. The boundary is restricted to LineStrings or CompositeCurves)
	OrientableSurface
	TexturedSurface (defined in CityGML’s *TexturedSurface* module, not in GML. This modelling approach has been marked deprecated)
	CompositeSurface (members are restricted to Orienta- bleSurfaces, TexturedSurfaces, Polygons or Compo- siteSurfaces)
	TriangulatedSurface
	TIN
_Curve	LineString
	CompositeCurve (members are restricted to LineStrings or CompositeCurves)
	Point
_Coverage	RectifiedGridCoverage
_AbstractGeometricAggregate	MultiSolid
	MultiSurface (members are restricted to OrientableSurfaces, TexturedSurfaces, Polygons or CompositeSurfaces)
	MultiCurve (members are restricted to LineStrings or CompositeCurves)
	MultiPoint
	GeometricComplex (restricted to connected, linear networks)

Annex E: （资料性附录）细节层次模型的特征分配概述

下表列出了CityGML的全部特征类型。对于每一个特征类型，表格提供了该类型所包含的全部非空间和空间属性。同时，每一个特征（feature）都对应着一个LOD精度区间，和相应精度所代表的属性（property）。每一个特征类型的名称都开始于一个表达该类型所在CityGML模块的前缀。关于CityGML模块和前缀关系的列表，请参考章节4.3和章节7。

Feature Class	Property	Type	LOD
CityGML Core module			
core:CityModelType			0 – 4
	cityObjectMember	gml:FeaturePropertyType	0 – 4
	app:appearanceMember	app:AppearancePropertyType	0 – 4
	_GenericApplicationProperty OfCityModel	xs:anyType	0 – 4
core:AbstractCityObjectType			0 – 4
	creationDate	xs:date	0 – 4
	terminationDate	xs:date	0 – 4
	externalReference	core:ExternalReferenceType	0 – 4
	generalizesTo	core:GeneralizationRelationType	0 – 4
	relativeToTerrain	core:RelativeToTerrainType	0 – 4
	relativeToWater	core:RelativeToWaterType	0 – 4
	app:appearance	app:AppearancePropertyType	0 – 4
	gen:_genericAttribute	gen:AbstractGenericAttributeType	0 – 4
	_GenericApplicationProperty OfCityObject	xs:anyType	0 – 4
core:AbstractSiteType			0 – 4
	_GenericApplicationProperty OfSite	xs:anyType	0 – 4

Feature Class	Property	Type	LOD
core:AddressType			0 – 4
	xalAddress	core:xalAddressPropertyType	0 – 4
	multiPoint	gml:MultiPointPropertyType	0 – 4
	_GenericApplicationPropertyOfAddress	xs:anyType	0 – 4
Appearance module			
app:AppearanceType			0 – 4
	theme	xs:string	0 – 4
	surfaceDataMember	app:SurfaceDataPropertyType	0 – 4
	_GenericApplicationPropertyOfAppearance	xs:anyType	0 – 4
app:AbstractSurfaceDataType			0 – 4
	isFront	xs:Boolean	0 – 4
	_GenericApplicationPropertyOfSurfaceData	xs:anyType	0 – 4
app:AbstractTextureType			0 – 4
	imageURI	xs:anyURI	0 – 4
	mimeType	gml:CodeType	0 – 4
	textureType	app:TextureTypeType	0 – 4
	wrapMode	app:WrapModeType	0 – 4
	borderColor	app:ColorPlusOpacity	0 – 4
	_GenericApplicationPropertyOfTexture	xs:anyType	0 – 4
app:ParameterizedTextureType			0 – 4
	target	app:TextureAssociationType	0 – 4
	_GenericApplicationPropertyOfParameterizedTexture	xs:anyType	0 – 4
app:GeoreferencedTextureType			0 – 4

Feature Class	Property	Type	LOD
	preferWorldFile	xs:boolean	0 – 4
	referencePoint	gml:PointPropertyType	0 – 4
	orientation	core:TransformationMatrix2x2Type	0 – 4
	target	xs:anyURI	0 – 4
	_GenericApplicationPropertyOfGeoreferencedTexture	xs:anyType	0 – 4
app:X3DMaterialType			0 – 4
	ambientIntensity	core:doubleBetween0and1	0 – 4
	diffuseColor	app:Color	0 – 4
	emissiveColor	app:Color	0 – 4
	specularColor	app:Color	0 – 4
	shininess	core:doubleBetween0and1	0 – 4
	transparency	core:doubleBetween0and1	0 – 4
	isSmooth	xs:boolean	0 – 4
	target	xs:anyURI	0 – 4
	_GenericApplicationProperty OfX3DMaterial	xs:anyType	0 – 4
Building module			
bldg:AbstractBuildingType			0 – 4

Feature Class	Property	Type	LOD

	boundedBy	bldg:BoundarySurfacePropertyType	2 – 4
Feature Class	**Property**	**Type**	**LOD**
	lod3Solid	gml:SolidPropertyType	3
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod3MultiCurve	gml:MultiCurvePropertyType	3
	lod3TerrainIntersection	gml:MultiCurvePropertyType	3
	lod4Solid	gml:SolidPropertyType	4
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod4MultiCurve	gml:MultiCurvePropertyType	4
	lod4TerrainIntersection	gml:MultiCurvePropertyType	4
	interiorRoom	bldg:InteriorRoomPropertyType	4
	consistsOfBuildingPart	bldg:BuildingPartPropertyType	0 – 4
	address	core:AddressPropertyType	0 – 4
	_GenericApplicationProperty OfAbstractBuilding	xs:anyType	0 – 4
bldg:BuildingType			0 – 4
	_GenericApplicationProperty OfBuiding	xs:anyType	0 – 4
bldg:BuildingPartType			0 – 4
	_GenericApplicationProperty OfBuidingPart	xs:anyType	0 – 4
bldg:BuildingInstallationType			2 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	2 – 4
	function	gml:CodeType	2 – 4
	usage	gml:CodeType	2 – 4
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	bldg:BoundarySurfacePropertyType	2 – 4
	_GenericApplicationProperty OfBuidingInstallation	xs:anyType	2 – 4
bldg:IntBuildingInstallationType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	bldg:BoundarySurfacePropertyType	4
	_GenericApplicationProperty OfIntBuidingInstallation	xs:anyType	4
bldg:AbstractBoundarySurface Type			2 – 4

Feature Class	Property	Type	LOD
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	opening	bldg:OpeningPropertyType	3 – 4
	_GenericApplicationProperty OfBoundarySurface	xs:anyType	2 – 4
bldg:RoofSurfaceType			2 – 4
	_GenericApplicationProperty OfRoofSurface	xs:anyType	2 – 4
bldg:WallSurfaceType			2 – 4
	_GenericApplicationProperty OfWallSurface	xs:anyType	2 – 4
bldg:OuterCeilingSurfaceType			2 – 4
	_GenericApplicationProperty OfOuterCeilingSurface	xs:anyType	2 – 4
bldg:OuterFloorSurfaceType			2 – 4
	_GenericApplicationProperty OfOuterFloorSurface	xs:anyType	2 – 4
bldg:GroundSurfaceType			2 – 4
	_GenericApplicationProperty OfGroundSurface	xs:anyType	2 – 4
bldg:ClosureSurfaceType			2 – 4
	_GenericApplicationProperty OfClosureSurface	xs:anyType	2 – 4
bldg:FloorSurfaceType			4
	_GenericApplicationProperty OfFloorSurface	xs:anyType	4
bldg:InteriorWallSurfaceType			4

Feature Class	Property	Type	LOD
	_GenericApplicationPropertyOfInteriorWallSurface	xs:anyType	4
bldg:CeilingSurfaceType			4
	_GenericApplicationProperty OfCeilingSurface	xs:anyType	4
bldg:AbstractOpeningType			3 – 4
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationProperty OfOpening	xs:anyType	3 – 4
bldg:WindowType			3 – 4
	_GenericApplicationProperty OfWindow	xs:anyType	3 – 4
bldg:DoorType			3 – 4
	address	core:AddressPropertyType	3 – 4
	_GenericApplicationProperty OfDoor	xs:anyType	3 – 4
bldg:RoomType			4

Feature Class	Property	Type	LOD
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Solid	gml:SolidPropertyType	4
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	boundedBy	bldg:BoundarySurfacePropertyType	4
	interiorFurniture	bldg:InteriorFurniturePropertyType	4
	roomInstallation	bldg:IntBuildingInstallationPropertyType	4
	_GenericApplicationProperty OfRoom	xs:anyType	4
bldg:BuildingFurnitureType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationProperty OfBuildingFurniture	xs:anyType	4
Bridge module			
brid:AbstractBridgeType			1 – 4

Feature Class	Property	Type	LOD

Feature Class	Property	Type	LOD
	lod4Solid	gml:SolidPropertyType	4
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod4MultiCurve	gml:MultiCurvePropertyType	4
	lod4TerrainIntersection	gml:MultiCurvePropertyType	4
	interiorBridgeRoom	brid:InteriorBridgeRoomPropertyType	4
	consistsOfBridgePart	brid:BridgePartPropertyType	1 – 4
	address	core:AddressPropertyType	1 – 4
	_GenericApplicationProperty OfAbstractBridge	xs:anyType	1 – 4
brid:BridgeType			1 – 4
	_GenericApplicationProperty OfBridge	xs:anyType	1 – 4
brid:BridgePartType			1 – 4
	_GenericApplicationProperty OfBridgePart	xs:anyType	1 – 4
brid:BridgeConstructionElement Type			2 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	lod1Geometry	gml:GeometryPropertyType	1
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod1TerrainIntersection	gml:MultiCurvePropertyType	1
	lod2TerrainIntersection	gml:MultiCurvePropertyType	2
	lod3TerrainIntersection	gml:MultiCurvePropertyType	3
	lod4TerrainIntersection	gml:MultiCurvePropertyType	4
	lod1ImplicitRepresentation	core:ImplicitRepresentationPropertyType	1
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	brid:BoundarySurfacePropertyType	2 – 4
	_GenericApplicationProperty OfBridgeConstruc-tionElement	xs:anyType	2 – 4
brid:BridgeInstallationType			2 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	2 – 4
	function	gml:CodeType	2 – 4
	usage	gml:CodeType	2 – 4
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	brid:BoundarySurfacePropertyType	2 – 4
	_GenericApplicationProperty OfBridgeInstallation	xs:anyType	2 – 4
brid:IntBridgeInstallationType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	brid:BoundarySurfacePropertyType	4
	_GenericApplicationProperty OfIntBridgeInstallation	xs:anyType	4
brid:AbstractBoundarySurface Type			2 – 4

Feature Class	Property	Type	LOD
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	opening	brid:OpeningPropertyType	3 – 4
	_GenericApplicationProperty OfBoundarySurface	xs:anyType	2 – 4
brid:RoofSurfaceType			2 – 4
	_GenericApplicationProperty OfRoofSurface	xs:anyType	2 – 4
brid:WallSurfaceType			2 – 4
	_GenericApplicationProperty OfWallSurface	xs:anyType	2 – 4
brid:OuterCeilingSurfaceType			2 – 4
	_GenericApplicationProperty OfOuterCeilingSurface	xs:anyType	2 – 4
brid:OuterFloorSurfaceType			2 – 4
	_GenericApplicationProperty OfOuterFloorSurface	xs:anyType	2 – 4
brid:GroundSurfaceType			2 – 4
	_GenericApplicationProperty OfGroundSurface	xs:anyType	2 – 4
brid:ClosureSurfaceType			2 – 4
	_GenericApplicationProperty OfClosureSurface	xs:anyType	2 – 4
brid:FloorSurfaceType			4
	_GenericApplicationProperty OfFloorSurface	xs:anyType	4
brid:InteriorWallSurfaceType			4

Feature Class	Property	Type	LOD
	_GenericApplicationPropertyOfInteriorWallSurface	xs:anyType	4
brid:CeilingSurfaceType			4
	_GenericApplicationProperty OfCeilingSurface	xs:anyType	4
brid:AbstractOpeningType			3 – 4
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationProperty OfOpening	xs:anyType	3 – 4
brid:WindowType			3 – 4
	_GenericApplicationProperty OfWindow	xs:anyType	3 – 4
brid:DoorType			3 – 4
	address	core:AddressPropertyType	3 – 4
	_GenericApplicationProperty OfDoor	xs:anyType	3 – 4
brid: BridgeRoomType			4

Feature Class	Property	Type	LOD
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Solid	gml:SolidPropertyType	4
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	boundedBy	brid:BoundarySurfacePropertyType	4
	interiorFurniture	brid:InteriorFurniturePropertyType	4
	bridgeRoomInstallation	brid:IntBridgeInstallationPropertyType	4
	_GenericApplicationProperty OfBridgeRoom	xs:anyType	4
brid: BridgeFurnitureType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationProperty OfBridgeFurniture	xs:anyType	4
Relief module			
dem:ReliefFeatureType			0 – 4
	lod	core:integerBetween0and4	0 – 4
	reliefComponent	dem:ReliefComponentPropertyType	0 – 4
	_GenericApplicationProperty OfReliefFeature	xs:anyType	0 – 4
dem:AbstractReliefComponent Type			0 – 4

Feature Class	Property	Type	LOD
	lod	core:integerBetween0and4	0 – 4
	extent	gml:PolygonPropertyType	0 – 4
	_GenericApplicationPropertyOfReliefComponent	xs:anyType	0 – 4
dem:TINReliefType			0 – 4
	tin	dem:tinPropertyType	0 – 4
	_GenericApplicationPropertyOfTINRelief	xs:anyType	0 – 4
dem:RasterReliefType			0 – 4
	grid	dem:gridPropertyType	0 – 4
	_GenericApplicationPropertyOfRasterRelief	xs:anyType	0 – 4
dem:MassPointReliefType			0 – 4
	reliefPoints	gml:MultiPointPropertyType	0 – 4
	_GenericApplicationPropertyOfMassPointRelief	xs:anyType	0 – 4
dem:BreakLineReliefType			0 – 4
	ridgeOrValleyLines	gml:MultiCurvePropertyType	0 – 4
	breaklines	gml:MultiCurvePropertyType	0 – 4
	_GenericApplicationPropertyOfBreakLineRelief	xs:anyType	0 – 4
CityFurniture module			
frn:CityFurnitureType			1 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	lod1Geometry	gml:GeometryPropertyType	1
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod1TerrainIntersection	gml:MultiCurvePropertyType	1
	lod2TerrainIntersection	gml:MultiCurvePropertyType	2
	lod3TerrainIntersection	gml:MultiCurvePropertyType	3
	lod4TerrainIntersection	gml:MultiCurvePropertyType	4
	lod1ImplicitRepresentation	core:ImplicitRepresentationPropertyType	1
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationPropertyOfCityFurniture	xs:anyType	1 – 4
CityObjectGroup module			
grp:CityObjectGroupType			0 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	0 – 4
	function	gml:CodeType	0 – 4
	usage	gml:CodeType	0 – 4
	groupMember	grp:CityObjectGroupMemberType	0 – 4
	parent	grp:CityObjectGroupParentType	0 – 4
	geometry	gml:GeometryPropertyType	0 – 4
	_GenericApplicationProperty OfCityObjectGroup	xs:anyType	0 – 4
Generics module			
gen:GenericCityObjectType			0 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	0 – 4
	function	gml:CodeType	0 – 4
	usage	gml:CodeType	0 – 4
	lod0Geometry	gml:GeometryPropertyType	0
	lod1Geometry	gml:GeometryPropertyType	1
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod0TerrainIntersection	gml:MultiCurvePropertyType	0
	lod1TerrainIntersection	gml:MultiCurvePropertyType	1
	lod2TerrainIntersection	gml:MultiCurvePropertyType	2
	lod3TerrainIntersection	gml:MultiCurvePropertyType	3
	lod4TerrainIntersection	gml:MultiCurvePropertyType	4
	lod0ImplicitRepresentation	core:ImplicitRepresentationPropertyType	0
	lod1ImplicitRepresentation	core:ImplicitRepresentationPropertyType	1
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
LandUse module			
luse:LandUseType			0 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	0 – 4
	function	gml:CodeType	0 – 4
	usage	gml:CodeType	0 – 4
	lod0MultiSurface	gml:MultiSurfacePropertyType	0
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	_GenericApplicationPropertyOfLandUse	xs:anyType	0 – 4
Transportation module			
tran:AbstractTransportationObject Type			0 – 4
	_GenericApplicationPropertyOfTransportationObject	xs:anyType	0 – 4
tran:TransportationComplexType			0 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	0 – 4
	function	gml:CodeType	0 – 4
	usage	gml:CodeType	0 – 4
	trafficArea	tran:TrafficAreaPropertyType	0 – 4
	auxilaryTrafficArea	tran:AuxilaryTrafficAreaPropertyType	0 – 4
	lod0Network	gml:GeometricComplexPropertyType	0
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	_GenericApplicationPropertyOfTransportationComplex	xs:anyType	0 – 4
tran:TrafficAreaType			1 – 4
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	surfaceMaterial	gml:CodeType	1 – 4
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	_GenericApplicationPropertyOfTrafficArea	xs:anyType	1 – 4

Feature Class	Property	Type	LOD
tran:AuxillaryTrafficAreaType			1 – 4
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	surfaceMaterial	gml:CodeType	1 – 4
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	_GenericApplicationPropertyOfAuxiliaryTrafficArea	xs:anyType	1 – 4
tran:TrackType			1 – 4
	_GenericApplicationProperty OfTrack	xs:anyType	1 – 4
tran:RoadType			1 – 4
	_GenericApplicationProperty OfRoad	xs:anyType	1 – 4
tran:RailwayType			1 – 4
	_GenericApplicationProperty OfRailway	xs:anyType	1 – 4
tran:SquareType			1 – 4
	_GenericApplicationProperty OfSquare	xs:anyType	1 – 4
Tunnel module			
tun:AbstractTunnelType			1 – 4

Feature Class	Property	Type	LOD

Feature Class	Property	Type	LOD
	lod4MultiCurve	gml:MultiCurveProperty	4
	lod4TerrainIntersection	gml:MultiCurveProperty Type	4
	interiorHollowSpace	tun:InteriorHollowSpace PropertyType	4
	consistsOfTunnelPart	tun:TunnelPartPropertyT ype	1 – 4
	_GenericApplicationProp erty OfAbstractTunnel	xs:anyType	1 – 4
tun:TunnelType			1 – 4
	_GenericApplicationProp erty OfTunnel	xs:anyType	1 – 4
tun:TunnelPartType			1 – 4
	_GenericApplicationProp erty OfTunnelPart	xs:anyType	1 – 4
tun:TunnelInstallationTy pe			2 – 4
	class	gml:CodeType	2 – 4
	function	gml:CodeType	2 – 4
	usage	gml:CodeType	2 – 4
	lod2Geometry	gml:GeometryPropertyT ype	2
	lod3Geometry	gml:GeometryPropertyT ype	3
	lod4Geometry	gml:GeometryPropertyT ype	4
	lod2ImplicitRepresentati on	core:ImplicitRepresentat ionPropertyType	2
	lod3ImplicitRepresentati on	core:ImplicitRepresentat ionPropertyType	3
	lod4ImplicitRepresentati on	core:ImplicitRepresentat ionPropertyType	4
	boundedBy	tun:BoundarySurfacePro pertyType	2 – 4
	_GenericApplicationProp erty OfTunnelInstallation	xs:anyType	2 – 4
tun:IntTunnelInstallation Type			4

Feature Class	Property	Type	LOD
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	boundedBy	tun:BoundarySurfacePropertyType	4
	_GenericApplicationProperty OfIntTunnelInstallation	xs:anyType	4
tun:AbstractBoundarySurface Type			2 – 4
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	opening	tun:OpeningPropertyType	3 – 4
	_GenericApplicationProperty OfBoundarySurface	xs:anyType	2 – 4
tun:RoofSurfaceType			2 – 4
	_GenericApplicationProperty OfRoofSurface	xs:anyType	2 – 4
tun:WallSurfaceType			2 – 4
	_GenericApplicationProperty OfWallSurface	xs:anyType	2 – 4
tun:OuterCeilingSurface Type			2 – 4
	_GenericApplicationProperty OfOuterCeilingSurface	xs:anyType	2 – 4
tun:OuterFloorSurfaceType			2 – 4

Feature Class	Property	Type	LOD
	_GenericApplicationPropertyOfOuterFloorSurface	xs:anyType	2 – 4
tun:GroundSurfaceType			2 – 4
	_GenericApplicationPropertyOfGroundSurface	xs:anyType	2 – 4
tun:ClosureSurfaceType			2 – 4
	_GenericApplicationPropertyOfClosureSurface	xs:anyType	2 – 4
tun:FloorSurfaceType			4
	_GenericApplicationPropertyOfFloorSurface	xs:anyType	4
tun:InteriorWallSurfaceType			4
	_GenericApplicationPropertyOfInteriorWallSurface	xs:anyType	4
tun:CeilingSurfaceType			4
	_GenericApplicationPropertyOfCeilingSurface	xs:anyType	4
tun:AbstractOpeningType			3 – 4
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationPropertyOfOpening	xs:anyType	3 – 4
tun:WindowType			3 – 4
	_GenericApplicationPropertyOfWindow	xs:anyType	3 – 4
tun:DoorType			3 – 4

Feature Class	Property	Type	LOD
	address	core:AddressPropertyType	3 – 4
	_GenericApplicationPropertyOfDoor	xs:anyType	3 – 4
tun:HollowSpaceType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Solid	gml:SolidPropertyType	4
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	boundedBy	tun:BoundarySurfacePropertyType	4
	interiorFurniture	tun:InteriorFurniturePropertyType	4
	hollowSpaceInstallation	tun:IntTunnelInstallationPropertyType	4
	_GenericApplicationPropertyOfHollowSpace	xs:anyType	4
tun:TunnelFurnitureType			4
	class	gml:CodeType	4
	function	gml:CodeType	4
	usage	gml:CodeType	4
	lod4Geometry	gml:GeometryPropertyType	4
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationPropertyOfTunnelFurniture	xs:anyType	4
Vegetation module			
veg:AbstractVegetationObject Type			1 – 4
	_GenericApplicationPropertyOfVegetationObject	xs:anyType	1 – 4
veg:PlantCoverType			1 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	averageHeight	gml:LengthType	1 – 4
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod2MultiSurface	gml:MultiSurfacePropertyType	2
	lod3MultiSurface	gml:MultiSurfacePropertyType	3
	lod4MultiSurface	gml:MultiSurfacePropertyType	4
	lod1MultiSolid	gml:MultiSolidPropertyType	1
	lod2MultiSolid	gml:MultiSolidPropertyType	2
	lod3MultiSolid	gml:MultiSolidPropertyType	3
	Lod4MultiSolid	gml:MultiSolidPropertyType	4
	_GenericApplicationProperty OfPlantCover	xs:anyType	1 – 4
veg:SolitaryVegetationObjectType			1 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	1 – 4
	function	gml:CodeType	1 – 4
	usage	gml:CodeType	1 – 4
	species	gml:CodeType	1 – 4
	height	gml:LengthType	1 – 4
	trunkDiameter	gml:LengthType	1 – 4
	crownDiameter	gml:LengthType	1 – 4
	lod1Geometry	gml:GeometryPropertyType	1
	lod2Geometry	gml:GeometryPropertyType	2
	lod3Geometry	gml:GeometryPropertyType	3
	lod4Geometry	gml:GeometryPropertyType	4
	lod1ImplicitRepresentation	core:ImplicitRepresentationPropertyType	1
	lod2ImplicitRepresentation	core:ImplicitRepresentationPropertyType	2
	lod3ImplicitRepresentation	core:ImplicitRepresentationPropertyType	3
	lod4ImplicitRepresentation	core:ImplicitRepresentationPropertyType	4
	_GenericApplicationProperty OfSolitaryVegetationObject	xs:anyType	1 – 4
WaterObject module			
wtr:AbstractWaterObjectType			0 – 4
	_GenericApplicationProperty OfWaterObject	xs:anyType	0 – 4
wtr:WaterBodyType			0 – 4

Feature Class	Property	Type	LOD
	class	gml:CodeType	0 – 4
	function	gml:CodeType	0 – 4
	usage	gml:CodeType	0 – 4
	lod0MultiCurve	gml:MultiCurvePropertyType	0
	lod1MultiCurve	gml:MultiCurvePropertyType	1
	lod0MultiSurface	gml:MultiSurfacePropertyType	0
	lod1MultiSurface	gml:MultiSurfacePropertyType	1
	lod1Solid	gml:SolidPropertyType	1
	lod2Solid	gml:SolidPropertyType	2
	lod3Solid	gml:SolidPropertyType	3
	lod4Solid	gml:SolidPropertyType	4
	boundedBy	wtr:BoundedByWaterSurface PropertyType	2 – 4
	_GenericApplicationProperty OfWaterBody	xs:anyType	0 – 4
wtr:AbstractWaterBoundary SurfaceType			2 – 4
	lod2Surface	gml:SurfacePropertyType	2
	lod3Surface	gml:SurfacePropertyType	3
	lod4Surface	gml:SurfacePropertyType	4
	_GenericApplicationProperty OfWaterBoundarySurface	xs:anyType	2 – 4
wtr:WaterSurfaceType			2 – 4
	waterLevel	WaterLevelType	2 – 4
	_GenericApplicationProperty OfWaterSurface	xs:anyType	2 – 4

Feature Class	Property	Type	LOD
wtr:WaterGroundSurface Type			2 – 4
	_GenericApplicationProp erty OfWaterGroundSurface	xs:anyType	2 – 4
wtr:WaterClosureSurface Type			2 – 4
	_GenericApplicationProp erty OfWaterClosureSurface	xs:anyType	2 – 4

Annex F: （资料性附录）CityGML 2.0变更日志

下表列出了CityGML 2.0 版本中新增或修改的所有特征类型、属性和数据结构。为了维护版本回溯的匹配性和稳定性，并没有删除CityGML 1.0 版本中所有标记为将弃用的类（classes）的相关代码。

Feature Class / Data Type	Property				Description of change
CityGML Core module					
core:AbstractCityObjectType					
	relativeToTerrain	x			new attribute
	relativeToWater	x			new attribute
Appearance module					
app:AbstractTextureType					
	mimeType		x		→ gml:CodeType
Building module					
bldg:AbstractBuildingType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
	roofType		x		→ gml:CodeType
	lod0FootPrint	x			new attribute
	lod0RoofEdge	x			new attribute
bldg:BuildingClassType				x	replaced by gml:CodeType
bldg:BuildingFunctionType				x	replaced by gml:CodeType
bldg:BuildingUsageType				x	replaced by gml:CodeType

Feature Class / Data Type	Property				Description of change
bldg:RoofTypeType				x	replaced by gml:CodeType
bldg:BuildingInstallationType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
	lod2ImplicitRepresentation	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
bldg:BuildingInstallation ClassType				x	replaced by gml:CodeType
bldg:BuildingInstallation FunctionType				x	replaced by gml:CodeType
bldg:BuildingInstallation UsageType				x	replaced by gml:CodeType
bldg:IntBuildingInstallationType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
	lod4ImplicitRepresentation		x		new attribute
	boundedBy		x		new attribute

Feature Class / Data Type	Property				Description of change
bldg:IntBuilding Installation ClassType				x	replaced by gml:CodeType
bldg:IntBuilding Installation				x	replaced by
FunctionType					gml:CodeType
bldg:IntBuilding Installation UsageType				x	replaced by gml:CodeType
bldg:OuterCeilingSurfaceType		x			new feature
	_GenericApplicationPropertyOfOuter	x			new attribute
	CeilingSurface				
bldg:OuterFloorSurfaceType		x			new feature
	_GenericApplicationPropertyOfOuter	x			new attribute
	FloorSurface				
bldg:AbstractOpeningType					
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
bldg:RoomType					
	class		x		àgml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
bldg:RoomClassType				x	replaced by gml:CodeType
bldg:RoomFunctionType				x	replaced by gml:CodeType

Feature Class / Data Type	Property				Description of change
bldg:RoomUsageType				x	replaced by gml:CodeType
bldg:BuildingFurnitureType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
bldg:BuildingFurnitureClassType				x	replaced by gml:CodeType
bldg:BuildingFurnitureFunctionType				x	replaced by gml:CodeType
bldg:BuildingFurnitureUsageType				x	replaced by gml:CodeType
Bridge module		x			New module
brid:AbstractBridgeType		x			new feature

Feature Class / Data Type	Property				Description of change

	section				
	lod4Solid	x			new attribute
	lod4MultiSurface	x			new attribute
	lod4MultiCurve	x			new attribute
	lod4TerrainIntersection	x			new attribute
	interiorBridgeRoom	x			new attribute
	consistsOfBridgePart	x			new attribute
	address	x			new attribute
	_GenericApplicationPropertyOfAbstract	x			new attribute
	Bridge				
brid:BridgeType		x			new feature
	_GenericApplicationPropertyOfBridge	x			new attribute
brid:BridgePartType		x			new feature
	_GenericApplicationPropertyOfBridge	x			new attribute
	Part				
brid:BridgeConstructionElementType		x			new feature

Feature Class / Data Type	**Property**				**Description of change**

Feature Class / Data Type	Property				Description of change
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod1Geometry	x			new attribute
	lod2Geometry	x			new attribute
	lod3Geometry	x			new attribute
	lod4Geometry	x			new attribute
	lod1TerrainIntersection	x			new attribute
	lod2TerrainIntersection	x			new attribute
	lod3TerrainIntersection	x			new attribute
	lod4TerrainIntersection	x			new attribute
	lod1ImplicitRepresentation	x			new attribute
	lod2ImplicitRepresentation	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
	_GenericApplicationPropertyOfBridge	x			new attribute
	ConstructionElement				
brid:BridgeInstallationType		x			new feature

Feature Class / Data Type	Property				Description of change
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod2Geometry	x			new attribute
	lod3Geometry	x			new attribute
	lod4Geometry	x			new attribute
	lod2ImplicitRepresentation	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
	_GenericApplicationPropertyOfBridge	x			new attribute
	Installation				
brid:IntBridgeInstallationType		x			new feature
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Geometry	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
	_GenericApplicationPropertyOfInt	x			new attribute
	BridgeInstallation				
brid:AbstractBoundarySurfaceType		x			new feature

Feature Class / Data Type	Property				Description of change
	lod2MultiSurface	x			new attribute
	lod3MultiSurface	x			new attribute
	lod4MultiSurface	x			new attribute
	opening	x			new attribute
	_GenericApplicationPropertyOf	x			new attribute
	BoundarySurface				
brid:RoofSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	RoofSurface				
brid:WallSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	WallSurface				
brid:OuterCeilingSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	OuterCeilingSurface				
brid:OuterFloorSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	OuterFloorSurface				
brid:GroundSurfaceType		x			new feature

Feature Class / Data Type	Property				Description of change
	_GenericApplicationPropertyOf	x			new attribute
	GroundSurface				
brid:ClosureSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	ClosureSurface				
brid:FloorSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	FloorSurface				
brid:InteriorWallSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	InteriorWallSurface				
brid:CeilingSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	CeilingSurface				
brid:AbstractOpeningType		x			new feature

Feature Class / Data Type	Property				Description of change
	lod3MultiSurface	x			new attribute
	lod4MultiSurface	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	_GenericApplicationPropertyOfOpening	x			new attribute
brid:WindowType		x			new feature
	_GenericApplicationPropertyOfWindow	x			new attribute
brid:DoorType		x			new feature
	address	x			new attribute
	_GenericApplicationPropertyOfDoor	x			new attribute
brid:BridgeRoomType		x			new feature

Feature Class / Data Type	Property				Description of change
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Solid	x			new attribute
	lod4MultiSurface	x			new attribute
	boundedBy	x			new attribute
	interiorFurniture	x			new attribute
	bridgeRoomInstallation	x			new attribute
	_GenericApplicationPropertyOf	x			new attribute
	BridgeRoom				
brid:BridgeFurnitureType		x			new feature
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Geometry	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	_GenericApplicationPropertyOfBridge	x			new attribute
	Furniture				
CityFurniture module					
frn:CityFurnitureType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage	x			new attribute

Feature Class / Data Type	Property				Description of change
frn:CityFurnitureClassType				x	replaced by gml:CodeType
frn:CityFurnitureFunctionType				x	replaced by gml:CodeType
CityObjectGroup module					
grp:CityObjectGroupType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
Generics module					
gen:GenericCityObjectType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
gen:GenericAttributeSetType		x			new data type
gen:MeasureAttributeType		x			new data type
LandUse module					
luse:LandUseType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType

Feature Class / Data Type	Property				Description of change
luse:LandUseClassType				x	replaced by gml:CodeType
luse:LandUseFunctionType				x	replaced by gml:CodeType
luse:LandUseUsageType				x	replaced by gml:CodeType
Transportation module					
tran:TransportationComplexType					
	class	x			
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
tran:TransportationComplexFunctionType				x	replaced by gml:CodeType
tran:TransportationComplexUsageType				x	replaced by gml:CodeType
tran:TrafficAreaType					
	class	x			new attribute
	function		x		→ gml:CodeType
	usage		x		à → gml:CodeType
	surfaceMaterial		x		à → gml:CodeType
tran:TrafficAreaFunctionType				x	replaced by gml:CodeType
tran:TrafficAreaUsageType				x	replaced by gml:CodeType
tran:TrafficSurfaceMaterialType				x	replaced by gml:CodeType

Feature Class / Data Type	Property				Description of change
tran:AuxillaryTrafficAreaType					
	class	x			new attribute
	function		x		→ gml:CodeType
	usage	x			new attribute
	surfaceMaterial		x		→ gml:CodeType
tran:AuxiliaryTrafficArea FunctionType				x	replaced by gml:CodeType
Tunnel module		x			New module
tun:AbstractTunnelType		x			new feature

Feature Class / Data Type	Property				Description of change

Feature Class / Data Type	Property	Description of change
	e	
	lod4MultiCurve	x			new attribute
	lod4TerrainInter section	x			new attribute
	interiorHollowS pace	x			new attribute
	consistsOfTunn elPart	x			new attribute
	_GenericApplica tionPropertyOfA bstract	x			new attribute
	Tunnel				
tun:TunnelType		x			new feature
	_GenericApplica tionPropertyOfT unnel	x			new attribute
tun:TunnelPartT ype		x			new feature
	_GenericApplica tionPropertyOfT unnel	x			new attribute
	Part				
tun:TunnelInsta llationType		x			new feature

Feature Class / Data Type	Property				Description of change
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod2Geometry	x			new attribute
	lod3Geometry	x			new attribute
	lod4Geometry	x			new attribute
	lod2ImplicitRepresentation	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
	_GenericApplicationPropertyOfTunnel	x			new attribute
	Installation				
tun:IntTunnelInstallationType		x			new feature
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Geometry	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	boundedBy	x			new attribute
	_GenericApplicationPropertyOfInt	x			new attribute
	TunnelInstallation				
tun:AbstractBoundarySurfaceType		x			new feature

Feature Class / Data Type	Property				Description of change
	lod2MultiSurface	x			new attribute
	lod3MultiSurface	x			new attribute
	lod4MultiSurface	x			new attribute
	opening	x			new attribute
	_GenericApplicationPropertyOf	x			new attribute
	BoundarySurface				
tun:RoofSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	RoofSurface				
tun:WallSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	WallSurface				
tun:OuterCeilingSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	OuterCeilingSurface				
tun:OuterFloorSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	OuterFloorSurface				
tun:GroundSurfaceType		x			new feature

Feature Class / Data Type	Property				Description of change
	_GenericApplicationPropertyOf	x			new attribute
	GroundSurface				
tun:ClosureSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	ClosureSurface				
tun:FloorSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	FloorSurface				
tun:InteriorWallSurfaceType		x			new feature
	_GenericApplicationPropertyOf	x			new attribute
	InteriorWallSurface				
tun:CeilingSurfaceType		x			new feature
	_GenericApplicationPropertyOfCeilingSurface	x			new attribute
tun:AbstractOpeningType		x			new feature
	lod3MultiSurface	x			new attribute
	lod4MultiSurface	x			new attribute
	lod3ImplicitRepresentation	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	_GenericApplicationPropertyOfOpening	x			new attribute

Feature Class / Data Type	Property				Description of change
tun:WindowType		x			new feature
	_GenericApplicationPropertyOfWindow	x			new attribute
tun:DoorType		x			new feature
	_GenericApplicationPropertyOfDoor	x			new attribute
tun:HollowSpaceType		x			new feature
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Solid	x			new attribute
	lod4MultiSurface	x			new attribute
	boundedBy	x			new attribute
	interiorFurniture	x			new attribute
	hollowSpaceInstallation	x			new attribute
	_GenericApplicationPropertyOf	x			new attribute
	HollowSpace				
tun:TunnelFurnitureType		x			new feature

Feature Class / Data Type	Property				Description of change
	class	x			new attribute
	function	x			new attribute
	usage	x			new attribute
	lod4Geometry	x			new attribute
	lod4ImplicitRepresentation	x			new attribute
	_GenericApplicationPropertyOf	x			new attribute
	TunnelFurniture				
Vegetation module					
veg:PlantCoverType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage	x			new attribute
	lod4MultiSolid	x			New Representation
veg:PlantCoverClassType				x	replaced by gml:CodeType
veg:PlantCoverFunctionType				x	replaced by gml:CodeType
veg:SolitaryVegetationObjectType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage	x			new attribute
	species		x		→ gml:CodeType
veg:PlantClassType				x	replaced by gml:CodeType

Feature Class / Data Type	Property				Description of change
veg:PlantFunctionType				x	replaced by gml:CodeType
veg:SpeciesType				x	replaced by gml:CodeType
WaterObject module					
wtr:WaterBodyType					
	class		x		→ gml:CodeType
	function		x		→ gml:CodeType
	usage		x		→ gml:CodeType
wtr:WaterBodyClassType				x	replaced by gml:CodeType
wtr:WaterBodyFunctionType				x	replaced by gml:CodeType
wtr:WaterBodyUsageType				x	replaced by gml:CodeType
wtr:WaterSurfaceType					
	waterLevel		x		→ gml:CodeType
wtr:WaterLevelType				x	replaced by gml:CodeType

Annex G: （资料性附录）CityGML数据集示例

此章节提供了一些CityGML模型的案例和相对应的CityGML实例文档的节选。章节G.1到章节G.6详细阐述了如何表达并修改一个CityGML单体建筑，使其从粗略的LOD0模型逐步形成包含建筑室内、语义丰富、几何拓扑结构完整的LOD4精度等级的模型。该模型示例了CityGML建筑模块中的概念，并演示了如何将语义信息和几何数据及拓扑属性进行衔接，建立完整连通的*Building*模型。G.7章节中，通过建筑物及其周围的地形更加丰富的外观信息，详细表达了CityGML *Appearance*模块的用法。在G.8章节中给出了*Appearance*模块的另一个示例，该示例展示了带有内部孔洞多边形的纹理效果。最后，在G.9章节中说明了在CityGML中使用本地工程坐标参考系统的详细情况。

本附件中包含的所有CityGML实例文档都与CityGML模式包绑定在一起，并可以在以下位置获得：http://schemas.opengis.net/citygml/examples/2.0/ 。

G.1. CityGML数据集示例：LOD0级别建筑物

第一个示例演示了一个处于由不规则三角形(*dem:TINRefief*)组成的LOD0级别地形模型之中的同精度CityGML建筑物模型。该建筑模型包含建筑基底轮廓线和屋顶边缘线(*bldg：lod0FootPrint*和*bldg：lod0RoofEdge*)。建筑基底和屋顶的几何数据以*gml：MultiSurface*表达，并且遵循*Building*模块第二项要求的规定（请参见第10.3.9节），即属于同一表面的所有3D坐标元组的高度值必须完全相同（构成水平面）。建筑基底平面（深蓝色）位于建筑物的最低处，而屋顶平面（浅蓝色）位于屋顶边缘高度，因此悬浮于地形上。同时，使用xAL地址架构为建筑物赋予地址信息。图75左侧显示了LOD0模型，右侧表达的是所包含语义实体的层次结构。

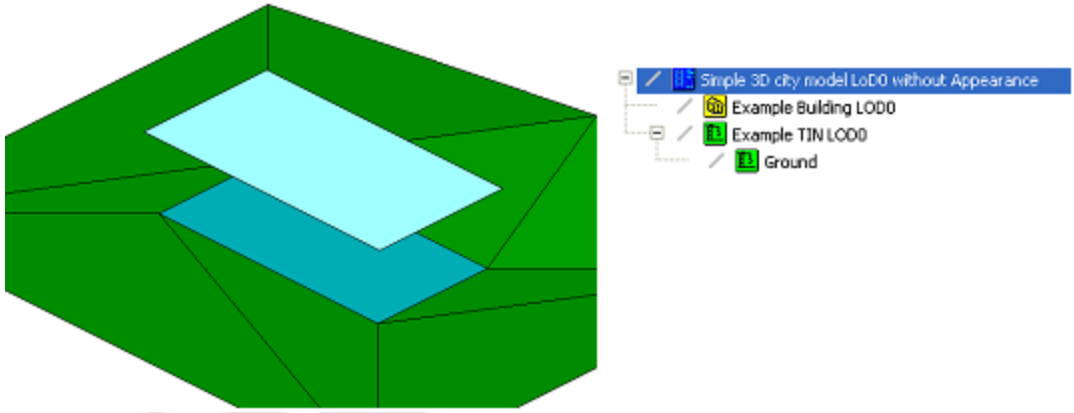

图 75. LOD0级别CityGML建筑模型（包含建筑基底轮廓和建筑屋顶轮廓）和地形模型的实例；右侧为模型层级结构。

清单3：摘自图 75 中可视化的LOD0级别建筑物的 CityGML 数据集。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
```

```
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:name>Simple 3D city model LOD0 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD0 </gml:name>
<bldg:function
codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
uilding_function.xml">1000</bldg:function>
<bldg:yearOfConstruction>1985</bldg:yearOfConstruction>
<bldg:roofType
codeSpace=http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBu
ilding_roofType.xml">1030</bldg:roofType>
<bldg:measuredHeight uom="#m">5.0</bldg:measuredHeight>
<bldg:storeysAboveGround>1</bldg:storeysAboveGround>
<bldg:storeyHeightsAboveGround uom="#m">
3.0</bldg:storeyHeightsAboveGround>
<bldg:lod0FootPrint>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458885.0 5438350.0 112.0 458885.0
5438355.0 112.0 458875.0
5438355.0 112.0 458875.0 5438350.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod0FootPrint>
<bldg:lod0RoofEdge>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458874.6 5438355.312347524 115.0 458874.6 5438349.687652476
115.0 458885.4 5438349.687652476
115.0 458885.4 5438355.312347524 115.0 458874.6 5438355.312347524 115.0
</gml:posList>
```

```
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod0RoofEdge>
<bldg:address>
<Address>
<xalAddress>
<xAL:AddressDetails>
<xAL:Country>
<xAL:CountryName>Germany</xAL:CountryName>
<xAL:Locality Type="Town">
<xAL:LocalityName>Eggenstein-Leopoldshafen</xAL:LocalityName>
<xAL:Thoroughfare Type="Street">
<xAL:ThoroughfareNumber>1</xAL:ThoroughfareNumber>
<xAL:ThoroughfareName>Hermann-von-Helmholtz-Platz</xAL:ThoroughfareName>
</xAL:Thoroughfare>
<xAL:PostalCode>
<xAL:PostalCodeNumber>76344</xAL:PostalCodeNumber>
</xAL:PostalCode>
</xAL:Locality>
</xAL:Country>
</xAL:AddressDetails>
</xalAddress>
<multiPoint>
<gml:MultiPoint>
<gml:pointMember>
<gml:Point>
<gml:pos srsDimension="3">458880.0 5438352.6 112.0 </gml:pos>
</gml:Point>
</gml:pointMember>
</gml:MultiPoint>
</multiPoint>
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
<gml:name>Example TIN LOD0</gml:name>
<dem:lod>0</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id="GUID_04D4DsNGv1MfvYu5O3lkcW">
<gml:name>Ground</gml:name>
<dem:lod>0</dem:lod>
<dem:tin>
<gml:TriangulatedSurface gml:id="ground">
<gml:trianglePatches>
```

```
<gml:Triangle>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458868 5438362 112 458875 5438355 112 458883 5438362 114
458868 5438362 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Triangle>
…
</gml:trianglePatches>
</gml:TriangulatedSurface>
</dem:tin>
</dem:TINRelief>
</dem:reliefComponent>
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

G.2. CityGML数据集示例：LOD1级别建筑物

图76是LOD1级别建筑物和地形模型的案例。在LOD1级别使用体块表达建筑物。建筑体块的几何形状由实体 (*gml:Solid*) 描述，其外壳由六个平面 (*gml:Polygon*) 界定。除了几何形状外，该建筑还增加了更多属性。其中一些属性是枚举属性，用于演示CityGML *code list*机制的用法（参见第 10.14 章）。编码属性值取自 SIG 3D 提议的代码列表（参见附件 C）。

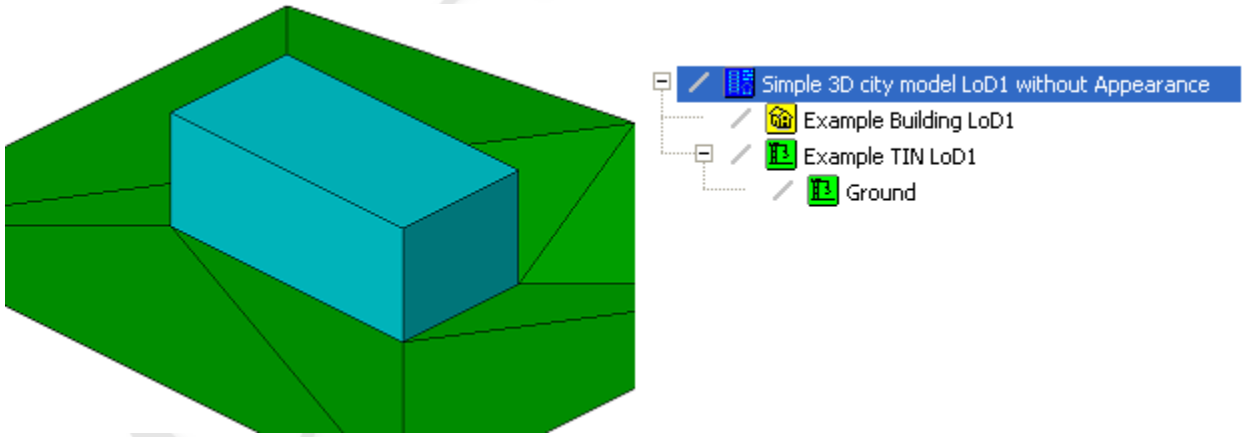

图 76. LOD1级别CityGML建筑模型示例（左侧为模型的三维表达；右侧为模型层级结构）。

清单 4：图 76 中可视化的LOD1级别建筑物的CityGML数据集摘录。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
```

```
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:name>Simple 3D city model LOD1 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD1 </gml:name>
<bldg:function
codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
uilding_function.xml">1000</bldg:function>
<bldg:yearOfConstruction>1985</bldg:yearOfConstruction>
<bldg:roofType
codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
uilding_roofType.xml">1030</bldg:roofType>
<bldg:measuredHeight uom="#m">5.0</bldg:measuredHeight>
<bldg:storeysAboveGround>1</bldg:storeysAboveGround>
<bldg:storeyHeightsAboveGround uom="#m">
3.0</bldg:storeyHeightsAboveGround>
<bldg:lod1Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface >
<!-- Face Side 1 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458885.0 5438350.0 112.0 458885.0
5438350.0 116.0 458875.0
5438350.0 116.0 458875.0 5438350.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 2 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458885.0 5438350.0 112.0 458885.0 5438355.0 112.0 458885.0
5438355.0 116.0 458885.0
```

```
5438350.0 116.0 458885.0 5438350.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 3 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458885.0 5438355.0 112.0 458875.0 5438355.0 112.0 458875.0
5438355.0 116.0 458885.0
5438355.0 116.0 458885.0 5438355.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 4 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438355.0 112.0 458875.0 5438350.0 112.0 458875.0
5438350.0 116.0 458875.0
5438355.0 116.0 458875.0 5438355.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Top -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 116.0 458885.0 5438350.0 116.0 458885.0
5438355.0 116.0 458875.0
5438355.0 116.0 458875.0 5438350.0 116.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Bottom -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458875.0 5438355.0 112.0 458885.0
5438355.0 112.0 458885.0
5438350.0 112.0 458875.0 5438350.0 112.0 </gml:posList>
```

```
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod1Solid>
<bldg:address>
<Address>
<xalAddress>
<xAL:AddressDetails>
<xAL:Country>
<xAL:CountryName>Germany</xAL:CountryName>
<xAL:Locality Type="Town">
<xAL:LocalityName>Eggenstein-Leopoldshafen</xAL:LocalityName>
<xAL:Thoroughfare Type="Street">
<xAL:ThoroughfareNumber>1</xAL:ThoroughfareNumber>
<xAL:ThoroughfareName>Hermann-von-Helmholtz-Platz</xAL:ThoroughfareName>
</xAL:Thoroughfare>
<xAL:PostalCode>
<xAL:PostalCodeNumber>76344</xAL:PostalCodeNumber>
</xAL:PostalCode>
</xAL:Locality>
</xAL:Country>
</xAL:AddressDetails>
</xalAddress>
<multiPoint>
<gml:MultiPoint>
<gml:pointMember>
<gml:Point>
<gml:pos srsDimension="3">458880.0 5438352.6 112.0 </gml:pos>
</gml:Point>
</gml:pointMember>
</gml:MultiPoint>
</multiPoint>
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
<gml:name>Example TIN LOD1</gml:name>
<dem:lod>1</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id="GML_4eb161b0-aa7e-4087-937c-5c4c427c7fc9">
<gml:name>Ground</gml:name>
<dem:lod>1</dem:lod>
<dem:tin>
```

```
<gml:TriangulatedSurface>
<gml:trianglePatches>
<gml:Triangle>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458868.0 5438362.0 112.0 458875.0 5438355.0 112.0 458883.0
5438362.0 114.0 458868.0
5438362.0 112.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Triangle>
<gml:Triangle>
…
</gml:Triangle>
… (more triangles)
</gml:trianglePatches>
</gml:TriangulatedSurface>
</dem:tin>
</dem:TINRelief>
</dem:reliefComponent>
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

G.3. CityGML数据集示例：LOD2级别建筑物

前几章的建筑模型现在以LOD2级别表示。该模型反映了建筑的屋顶结构并包含了构成模型边界的所有平面(*bldg:boundedBy*)，这些平面在语义上对建筑模型的外壳进行了分类（*bldg:RoofSurface*、*bldg:WallSurface* 和 *bldg:GroundSurface*）。除了带有属性的边界平面之外，该模型使用LOD2级别的实体 (*gml:Solid*) 描述建筑几何形态。根据*Building*模块（参见第 10.3.9 章）中第4项要求规定，该实体几何体块必须使用GML3 XLink机制（*xlink:href*，下面清单5中的粗体代码）来引用边界平面的几何数据。

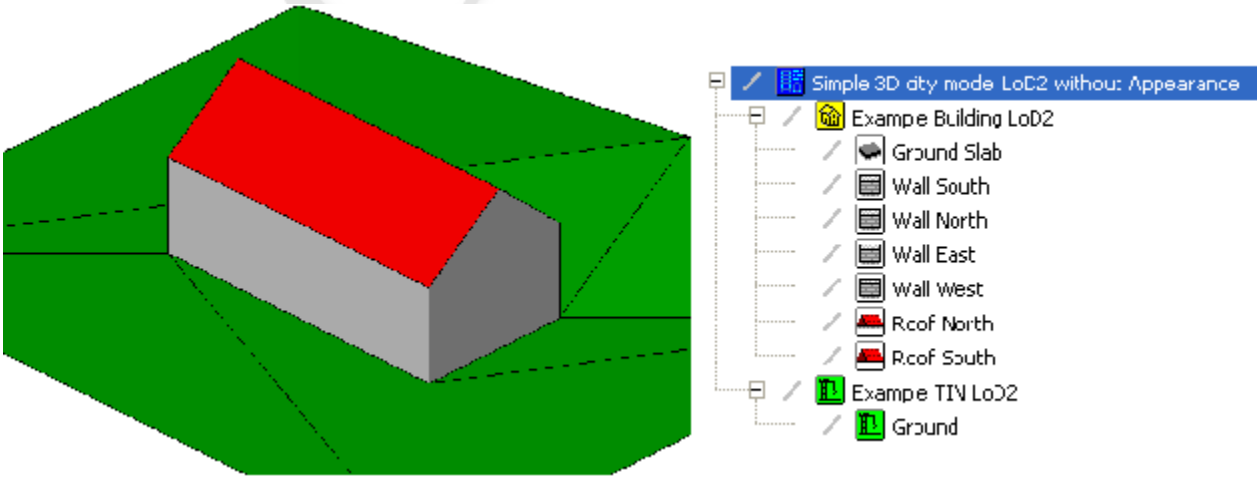

图 77. LOD2级别CityGML建筑模型示例（左侧为模型的三维表达；右侧为模型层级结构）。

清单 5：图 77 中可视化的LOD2级别建筑物的CityGML数据集摘录。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:name>Simple 3D city model LOD2 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD2 </gml:name>
…   (further attributes see LOD1 example)
<bldg:lod2Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<!-- Ground Slab -->
<gml:surfaceMember xlink:href="#GML_d3981803-d4b0-4b5b-969c-53f657594757"
/>
<!-- Wall South -->
<gml:surfaceMember xlink:href="#GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1"
/>
<!-- Wall North -->
<gml:surfaceMember xlink:href="#GML_d3909000-2f18-4472-8886-1c127ea67df1"
/>
<!-- Wall East -->
<gml:surfaceMember xlink:href="#GML_6286ffa9-3811-4796-a92f-3fd037c8e668"
/>
<!-- Wall West -->
<gml:surfaceMember xlink:href="#GML_5cc4fd92-d5de-4dd8-971e-892c91da2d9f"
/>
<!-- Roof North -->
<gml:surfaceMember xlink:href="#GML_ec6a8966-58d9-4894-8edd-9aceb91b923f"
/>
<!-- Roof South -->
<gml:surfaceMember xlink:href="#GML_b41dc792-5da6-4cd9-8f85-247583f305e3"
/>
</gml:CompositeSurface>
```

```
</gml:exterior>
</gml:Solid>
</bldg:lod2Solid>
<bldg:boundedBy>
<bldg:GroundSurface>
<gml:name>Ground Slab</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_d3981803-d4b0-4b5b-969c-53f657594757">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458875.0 5438355.0 112.0 458885.0
5438355.0 112.0 458885.0 5438350.0 112.0 458875.0 5438350.0 112.0
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:GroundSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall South</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458885.0 5438350.0 112.0 458885.0
5438350.0 115.0 458875.0 5438350.0 115.0 458875.0 5438350.0 112.0
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall North</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
```

```
<gml:Polygon gml:id="GML_d3909000-2f18-4472-8886-1c127ea67df1">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall East</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_6286ffa9-3811-4796-a92f-3fd037c8e668">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall West</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_5cc4fd92-d5de-4dd8-971e-892c91da2d9f">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:RoofSurface>
<gml:name>Roof North</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_ec6a8966-58d9-4894-8edd-9aceb91b923f">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:RoofSurface>
```

```
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:RoofSurface>
<gml:name>Roof South</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_b41dc792-5da6-4cd9-8f85-247583f305e3">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:RoofSurface>
</bldg:boundedBy>
<bldg:address>
<Address>
<xalAddress>
<xAL:AddressDetails>
…
</xAL:AddressDetails>
</xalAddress>
<multiPoint>
<gml:MultiPoint>
<gml:pointMember>
<gml:Point>
<gml:pos srsDimension="3">458880.0 5438352.7 112.0 </gml:pos>
</gml:Point>
</gml:pointMember>
</gml:MultiPoint>
</multiPoint>
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
<gml:name>Example TIN LOD2</gml:name>
<dem:lod>2</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id="GML_4eb161b0-aa7e-4087-937c-5c4c427c7fc9">
<gml:name>Ground</gml:name>
<dem:lod>2</dem:lod>
<dem:tin>
…
</dem:tin>
</dem:TINRelief>
</dem:reliefComponent>
</dem:ReliefFeature>
```

```
</cityObjectMember>
</CityModel>
```

G.4. LOD2级别建筑物的CityGML数据集示例: 具有CityGML拓扑特征的相邻建筑

这个例子说明了如何使用GML3的 XLink 机制（参见第 8.1 章）表达CityGML的拓扑结构。此示例的建筑模型是通过在LOD2建筑模型旁边增加一个相邻的车库得到的（参见图 78）。此车库在建模过程中被定义为建筑零件 (*bldg:BuildingPart*) 并与建筑外壳共享同一个平面几何图形。该模型对建筑物和车库都分别提供了边界平面 (*bldg:boundedBy*) 以及实体几何图形。车库与建筑物接触的墙面平面如图79所示。对于建筑物，该共用墙面的几何平面被拆分为非共享部分和共享部分（参见 *bldg:WallSurface* 和 *gml:name* “Wall East”）。车库通过其相应的 *bldg:WallSurface*（*gml:name*“Garage Wall West”）及其实体几何体来表达和建筑物东侧墙体接触的部分。由于需要对车库模型中与建筑接触部分的墙体平面进行反向， XLink的引用需要包含在 *gml:OrientableSurface* 元素中。这个XLink通过从车库指向建筑物几何图形来明确表示两个物体之间的拓扑邻接关系。

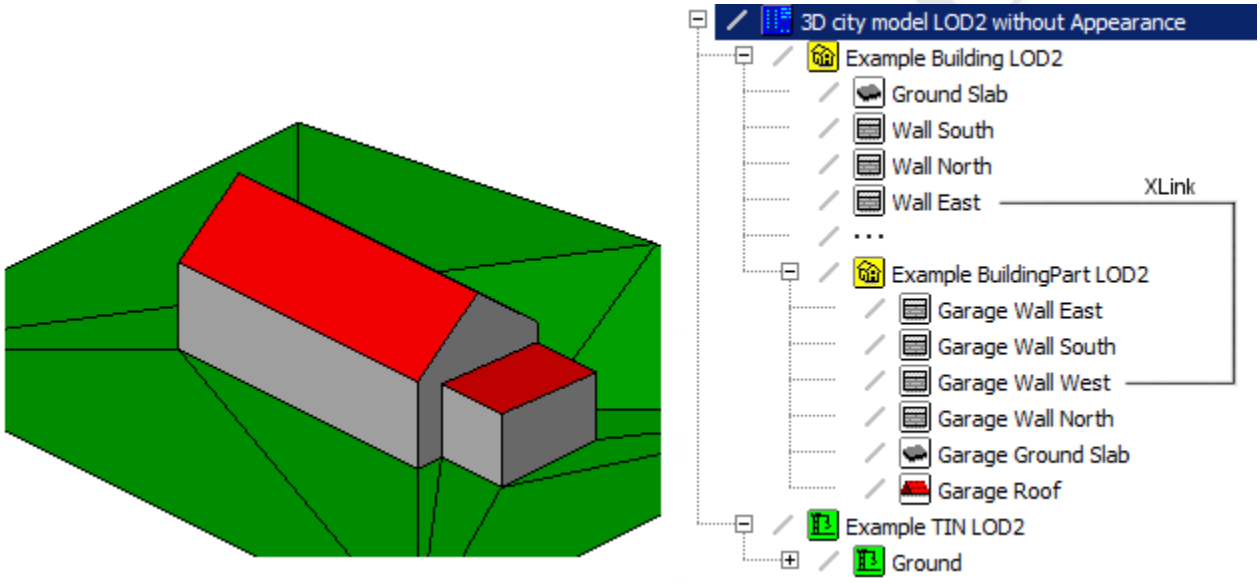

图 78. LOD2级别CityGML建筑模型示例（左侧为模型的三维表达；右侧为模型层级结构）。

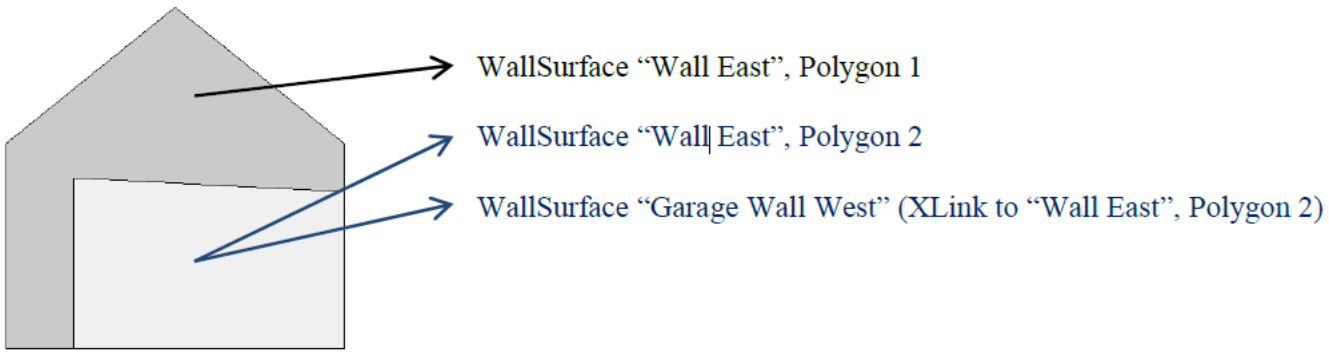

图 79. 车库与建筑物相接的边界平面。对于建筑物，该共用墙面的几何平面被拆分为非共享部分和共享部分（参见 bldg:WallSurface 和 gml:name “Wall East”）。车库通过其相应的 XLink机制来表达和建筑物东侧墙体接触的部分。

清单 6：摘自CityGML数据集的LOD2级别具有相邻车库的建筑物，如图 78 所示。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:dem="http://www.opengis.net/citygml/relief/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd">
<gml:name>3D city model LOD2 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD2</gml:name>
…   (further attributes see LOD1 example)
<bldg:lod2Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<!-- Ground Slab -->
<gml:surfaceMember xlink:href="#GML_d3981803-d4b0-4b5b-969c-53f657594757
"/>
<!-- Wall South -->
<gml:surfaceMember xlink:href="#GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1
"/>
<!-- Wall North -->
<gml:surfaceMember xlink:href="#GML_d3909000-2f18-4472-8886-1c127ea67df1
"/>
<!-- Wall East 1 -->
<gml:surfaceMember xlink:href="#GML_56d1dd88-36dd-4d1e-bff0-3305fbffa778
"/>
<!-- Wall East 2 -->
<gml:surfaceMember xlink:href="#GML_9f0465e6-f316-4f89-a9bd-eb21934ffe84
"/>
<!-- Wall West -->
<gml:surfaceMember xlink:href="#GML_5cc4fd92-d5de-4dd8-971e-892c91da2d9f
"/>
<!-- Roof North -->
<gml:surfaceMember xlink:href="#GML_ec6a8966-58d9-4894-8edd-9aceb91b923f
```

```
"/>
<!-- Roof South -->
<gml:surfaceMember xlink:href="#GML_b41dc792-5da6-4cd9-8f85-247583f305e3
"/>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod2Solid>
<bldg:boundedBy>
<bldg:GroundSurface>
<gml:name>Ground Slab</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_d3981803-d4b0-4b5b-969c-53f657594757">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458875.0 5438355.0 112.0 458885.0
5438355.0 112.0 458885.0 5438350.0 112.0 458875.0 5438350.0 112.0
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:GroundSurface>
</bldg:boundedBy>
…
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall East</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_56d1dd88-36dd-4d1e-bff0-3305fbffa778">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458885.0 5438350.0 112.0 458885.0 5438351.0 112.0 458885.0
5438351.0 114.5 458885.0 5438355.0 114.3 458885.0 5438355.0 115.0 458885.0
5438352.5 117.0 458885.0 5438350.0 115.0 458885.0 5438350.0
112.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_9f0465e6-f316-4f89-a9bd-eb21934ffe84">
<gml:exterior>
```

```
<gml:LinearRing>
<gml:posList>458885.0 5438355.0 112.0 458885.0 5438355.0 114.3 458885.0
5438351.0 114.5 458885.0 5438351.0 112.0 458885.0 5438355.0
112.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
…
<bldg:consistsOfBuildingPart>
<bldg:BuildingPart gml:id="GMLID_BUI379228_1244_301">
<gml:name>Example BuildingPart LOD2</gml:name>
<bldg:function codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_fu
nction.xml">1630</bldg:function>
<bldg:yearOfConstruction>1985</bldg:yearOfConstruction>
<bldg:roofType codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_ro
ofType.xml">1010</bldg:roofType>
<bldg:measuredHeight uom="#m">2.5</bldg:measuredHeight>
<bldg:lod2Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<!-- Garage Ground Slab -->
<gml:surfaceMember xlink:href="#GML_2e1ff653-b62b-41ee-9f99-d6852ae7d567
"/>
<!-- Garage Wall South -->
<gml:surfaceMember xlink:href="#GML_f3f56c7b-7e59-47bc-ba03-d841032f1a37
"/>
<!-- Garage Wall North -->
<gml:surfaceMember xlink:href="#GML_5339468c-b2cb-4a99-9eb5-8b0660fb26d3
"/>
<!-- Garage Wall East -->
<gml:surfaceMember xlink:href="#GML_dab75f49-f6f8-4490-b86b-450b613e1fc2
"/>
<!-- Garage Wall West (identical with Wall East 2 of Building) -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_9f0465e6-f316-4f89-a9bd-eb21934ffe84"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Garage Roof -->
<gml:surfaceMember xlink:href="#GML_7996bef1-f045-4704-be27-db27430d4f70
"/>
```

```
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod2Solid>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Garage Wall East</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_dab75f49-f6f8-4490-b86b-450b613e1fc2">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458887.5 5438355.0 114.3 458887.5 5438351.0 114.5 458887.5
5438351.0 112.0 458887.5
5438355.0 112.0 458887.5 5438355.0 114.3</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
…
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Garage Wall West</gml:name>
<bldg:lod2MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<!-- identical with Wall East 2 of Building -->
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_9f0465e6-f316-4f89-a9bd-eb21934ffe84"/>
</gml:OrientableSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod2MultiSurface>
</bldg:WallSurface>
</bldg:boundedBy>
…
</bldg:BuildingPart>
</bldg:consistsOfBuildingPart>
<bldg:address>
<Address>
<xalAddress>
<xAL:AddressDetails>
…
</xAL:AddressDetails>
```

```
</xalAddress>
<multiPoint>
…
</multiPoint>
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
<gml:name>Example TIN LOD2</gml:name>
<dem:lod>2</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id="GML_4eb161b0-aa7e-4087-937c-5c4c427c7fc9">
<gml:name>Ground</gml:name>
<dem:lod>2</dem:lod>
<dem:tin>
…
</dem:tin>
</dem:TINRelief>
</dem:reliefComponent>
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

G.5. CityGML数据集示例：LOD3级别建筑物

本章示例的LOD3级别建筑模型（参见图 80）将门 (*bldg:Door*)、窗户 (*bldg:Window*) 和屋顶外檐 (*bldg:RoofSurface*) 添加到前几章的展示的LOD2模型中。同样，LOD3精度级别建筑模型的实体几何图形是通过使用 GML3 XLink 机制 (*xlink:href*) 引用主题边界表面的几何图形实现的。为了获得有效的实体几何图形，屋顶面在几何上被分割为屋顶板和屋顶外檐部分。建筑实体几何图形仅参考屋顶板部分。由多个表面组成的墙被建模为 *gml:CompositeSurface*，然后被建筑实体引用。包含门窗开口的边界表面用具有一个外部和多个内部线性环的多边形进行建模（根据*Building*模块第8号一致性要求，参见第 10.4.8 章）。

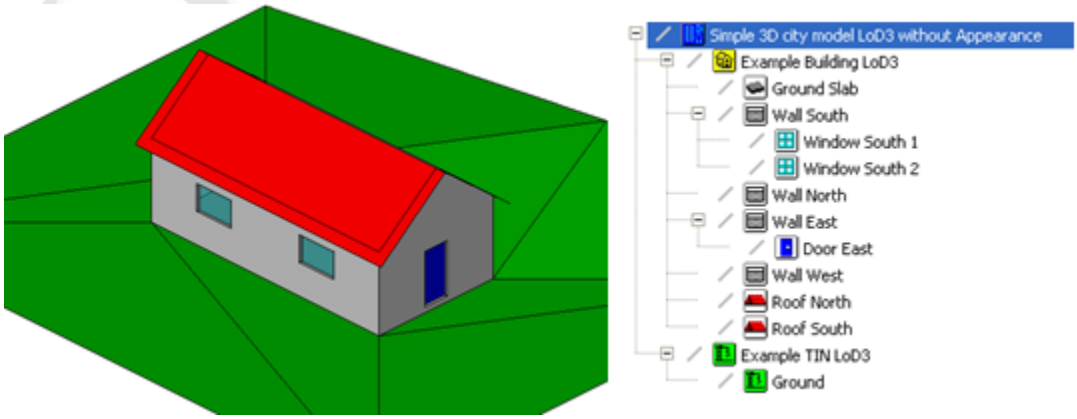

图 80. LOD3级别CityGML建筑模型示例（左侧为模型的三维表达；右侧为模型层级结构）。

清单 7：图 80 中可视化的LOD3级别建筑物的CityGML数据集摘录。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:name>Simple 3D city model LOD3 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD3 </gml:name>
…   (further attributes see LOD1 example)
<bldg:boundedBy>
<bldg:GroundSurface>
<gml:name>Ground Slab</gml:name>
…   (see LOD2 example)
</bldg:GroundSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall South</gml:name>
<bldg:lod3MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:CompositeSurface gml:id="GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1">
<gml:surfaceMember>
<gml:Polygon gml:id="PolyID10204_1916_571790_369478">
…
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="PolyID10205_105_876837_53833">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458875.0 5438350.0 112.0 458885.0 5438350.0 112.0 458885.0
5438350.0 115.0 458875.0 5438350.0 115.0 458875.0 5438350.0 112.0
</gml:posList>
</gml:LinearRing>
```

```
</gml:exterior>
<gml:interior>
<gml:LinearRing>
<gml:posList>458877.0 5438350.0 114.2 458878.5 5438350.0 114.2 458878.5
5438350.0 113.2 458877.0 5438350.0 113.2 458877.0 5438350.0 114.2
</gml:posList>
</gml:LinearRing>
</gml:interior>
<gml:interior>
<gml:LinearRing>
<gml:posList>458881.5 5438350.0 114.2 458883.0 5438350.0 114.2 458883.0
5438350.0 113.2 458881.5 5438350.0 113.2 458881.5 5438350.0 114.2
</gml:posList>
</gml:LinearRing>
</gml:interior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
…   (more surface members of the WallSurface)
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
<bldg:opening>
<bldg:Window gml:id="GML_3b09d6a5-4c24-4847-a8a2-e97475e3de47">
<gml:name>Window South 1</gml:name>
<bldg:lod3MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_5e07e2cc-c28c-480e-880f-dfdfe287bb9e">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
</bldg:Window>
</bldg:opening>
<bldg:opening>
<bldg:Window gml:id="GML_f75f01cc-c584-4a62-b34a-4a0e2640550d">
<gml:name>Window South 2</gml:name>
<bldg:lod3MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_d0ea2b6b-7992-4284-9a20-957a6c5c1cea">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
```

```
</bldg:lod3MultiSurface>
</bldg:Window>
</bldg:opening>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall North</gml:name>
…   (see LOD2 example)
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall East</gml:name>
<bldg:lod3MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:CompositeSurface gml:id="GML_6286ffa9-3811-4796-a92f-3fd037c8e668">
...
</gml:CompositeSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
<bldg:opening>
<bldg:Door gml:id="GML_93096bbb-5155-47fb-ae2c-e2f9327f3007">
<gml:name>Door East</gml:name>
<bldg:lod3MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_8f988da9-22d7-41e5-ae94-880afd46a3c9">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
</bldg:Door>
</bldg:opening>
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:WallSurface>
<gml:name>Wall West</gml:name>
…   (see LOD2 example)
</bldg:WallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:RoofSurface>
<gml:name>Roof North</gml:name>
<bldg:lod3MultiSurface>
```

```
<gml:MultiSurface>
<!-- Roof slab -->
<gml:surfaceMember>
<gml:Polygon gml:id="GML_ec6a8966-58d9-4894-8edd-9aceb91b923f">
…   (see LOD2 example)
</gml:Polygon>
</gml:surfaceMember>
<!-- Roof overhanging -->
<gml:surfaceMember>
<gml:Polygon gml:id="GML_70fa738e-80a4-4774-8a3b-322f037fa482">
<gml:exterior>
<gml:LinearRing>
<gml:posList>458874.6 5438352.5 117 458875 5438352.5 117 458875 5438355
115 458885 5438355 115 458885 5438352.5 117 458885.4 5438352.5 117
458885.4 5438355.312347524 114.75012198097823 458874.6 5438355.312347524
114.75012198097823 458874.6 5438352.5 117 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
</bldg:RoofSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:RoofSurface>
<gml:name>Roof South</gml:name>
<bldg:lod3MultiSurface>
<!-- Roof slab -->
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_b41dc792-5da6-4cd9-8f85-247583f305e3">
…   (see LOD2 example)
</gml:Polygon>
</gml:surfaceMember>
<!-- Roof overhanging -->
<gml:surfaceMember>
<gml:Polygon gml:id="GML_db6d8edc-4870-4523-a606-d440f36f8ec8">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod3MultiSurface>
</bldg:RoofSurface>
</bldg:boundedBy>
<bldg:lod3Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
```

```
<!-- Ground Slab -->
<gml:surfaceMember xlink:href="#GML_d3981803-d4b0-4b5b-969c-53f657594757
"/>
<!-- Wall South -->
<gml:surfaceMember xlink:href="#GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1
"/>
<!-- Window South 1 -->
<gml:surfaceMember xlink:href="#GML_5e07e2cc-c28c-480e-880f-dfdfe287bb9e
"/>
<!-- Window South 2 -->
<gml:surfaceMember xlink:href="#GML_d0ea2b6b-7992-4284-9a20-957a6c5c1cea
"/>
<!-- Wall North -->
<gml:surfaceMember xlink:href="#GML_d3909000-2f18-4472-8886-1c127ea67df1
"/>
<!-- Wall East -->
<gml:surfaceMember xlink:href="#GML_6286ffa9-3811-4796-a92f-3fd037c8e668
"/>
<!-- Door East -->
<gml:surfaceMember xlink:href="#GML_8f988da9-22d7-41e5-ae94-880afd46a3c9
"/>
<!-- Wall West -->
<gml:surfaceMember xlink:href="#GML_5cc4fd92-d5de-4dd8-971e-892c91da2d9f
"/>
<!-- Roof Slab North -->
<gml:surfaceMember xlink:href="#GML_ec6a8966-58d9-4894-8edd-9aceb91b923f
"/>
<!-- Roof Slab South -->
<gml:surfaceMember xlink:href="#GML_b41dc792-5da6-4cd9-8f85-247583f305e3
"/>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod3Solid>
<bldg:address>
<Address>
...   (see LOD1 example)
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
…   (see LOD1 example)
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

G.6. CityGML数据集示例：LOD4级别建筑物

在LOD4级别模型中，增加了建筑内部的内容（参见图 81）。该模型包含一个房间 (*bldg:Room*)，其中配备了一把摇椅 (*bldg:BuildingFurniture*)。房间以建筑内部的表面为边界进行围合（*bldg:InteriorWallSurface*、*bldg:FloorSurface*、*bldg:CeilingSurface*，通过房间的*bldg:boundedBy*属性关联），其几何图形由房间的 LOD4 实体几何图形 (*xlink:href*) 引用。如果内部边界表面的法向量指向房间，则在实体引用时，必须使用可定向表面 (*gml:OrientableSurface*) 翻转其方向，从而创建有效的实体几何图形（对于*gml:Solid*，包围体表面的法向量必须指向实体外）。

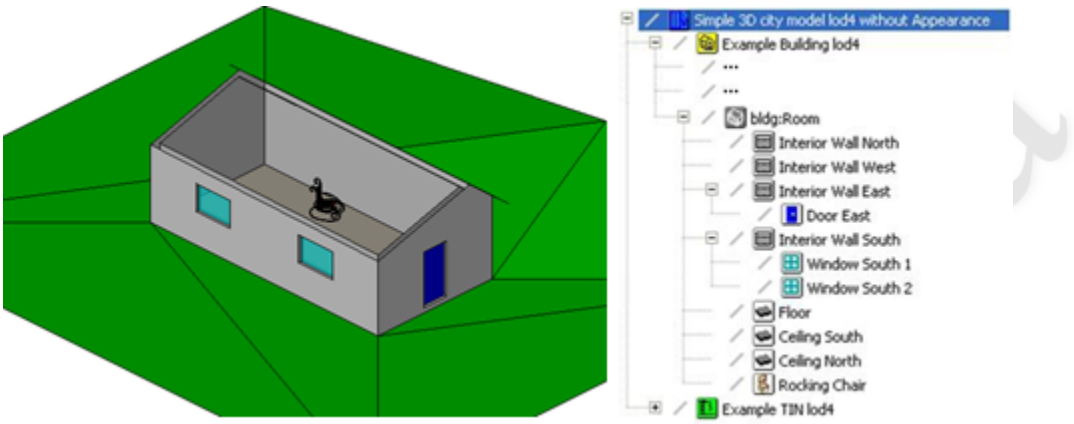

图 81. LOD4级别的CityGML建筑模型示例。为了更清晰地表达建筑室内和家具，建筑屋顶已被隐藏（左侧为模型的三维表达；右侧为模型层级结构）。

清单 8：图 81 中可视化的LOD4级别建筑物的CityGML数据集摘录。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:gml="
http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:name>Simple 3D city model LOD4 without Appearance</gml:name>
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD4 </gml:name>
…   (further attributes see LOD1 example)
```

```
<bldg:boundedBy>
…   (outer shell see LOD3 example)
<bldg:lod4Solid>
…   (building solid representation see LOD3 example)
</bldg:lod4Solid>
<bldg:interiorRoom>
<bldg:Room>
<bldg:lod4Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<!-- Floor -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_fa89e511-39b2-46de-9a13-9f4621576a46"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Interior Wall North -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_592ce9fa-0b98-4225-8d22-20eff4f86fc5"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Interior Wall West -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_a9fe597d-c338-43ad-a633-2a0beb273fac"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Interior Wall East -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_eaf1db16-56a3-4b86-ae19-2edbb604636f"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Door East -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="+">
<gml:baseSurface xlink:href="#GML_8f988da9-22d7-41e5-ae94-880afd46a3c9"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Interior Wall South -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_a718c157-c948-42cf-a786-0ce61044cff9"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Window South 1 -->
<gml:surfaceMember>
```

```
<gml:OrientableSurface orientation="+">
<gml:baseSurface xlink:href="#GML_5e07e2cc-c28c-480e-880f-dfdfe287bb9e"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Window South 2 -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="+">
<gml:baseSurface xlink:href="#GML_d0ea2b6b-7992-4284-9a20-957a6c5c1cea"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Ceiling North -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_989aa5cf-ee07-4fd8-89b6-500a9d5ba8041"/>
</gml:OrientableSurface>
</gml:surfaceMember>
<!-- Ceiling South -->
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_98841838-ee0b-402f-ba28-64ed61cb10f8"/>
</gml:OrientableSurface>
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod4Solid>
<bldg:boundedBy>
<bldg:InteriorWallSurface>
<gml:name>Interior Wall North</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_592ce9fa-0b98-4225-8d22-20eff4f86fc5">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:InteriorWallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:InteriorWallSurface>
<gml:name>Interior Wall West</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_a9fe597d-c338-43ad-a633-2a0beb273fac">
…
</gml:Polygon>
```

```
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:InteriorWallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:InteriorWallSurface>
<gml:name>Interior Wall East</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:CompositeSurface gml:id="GML_eaf1db16-56a3-4b86-ae19-2edbb604636f">
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
<bldg:opening>
<bldg:Door>
<gml:name>Door East</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_8f988da9-22d7-41e5-ae94-880afd46a3c9">
</gml:baseSurface>
</gml:OrientableSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:Door>
</bldg:opening>
</bldg:InteriorWallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:InteriorWallSurface>
<gml:name>Interior Wall South</gml:name>
<bldg:lod4MultiSurface>
```

```
<gml:MultiSurface>
<gml:surfaceMember>
<gml:CompositeSurface gml:id="GML_a718c157-c948-42cf-a786-0ce61044cff9">
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_cf0b79ba-f31f-4bae-a10f-5bcc85ce2cf6">
<gml:exterior>
…
</gml:exterior>
<gml:interior>
…
</gml:interior>
<gml:interior>
…
</gml:interior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
<gml:surfaceMember>
…
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
<bldg:opening>
<bldg:Window>
<gml:name>Window South 1</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
```

```
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_5e07e2cc-c28c-480e-880f-dfdfe287bb9e">
</gml:baseSurface>
</gml:OrientableSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:Window>
</bldg:opening>
<bldg:opening>
<bldg:Window>
<gml:name>Window South 2</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:OrientableSurface orientation="-">
<gml:baseSurface xlink:href="#GML_d0ea2b6b-7992-4284-9a20-957a6c5c1cea">
</gml:baseSurface>
</gml:OrientableSurface>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:Window>
</bldg:opening>
</bldg:InteriorWallSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:FloorSurface>
<gml:name>Floor</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_fa89e511-39b2-46de-9a13-9f4621576a46">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:FloorSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:CeilingSurface>
<gml:name>Ceiling South</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_989aa5cf-ee07-4fd8-89b6-500a9d5ba8041">
…
```

```
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:CeilingSurface>
</bldg:boundedBy>
<bldg:boundedBy>
<bldg:CeilingSurface>
<gml:name>Ceiling North</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon gml:id="GML_98841838-ee0b-402f-ba28-64ed61cb10f8">
…
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
</bldg:CeilingSurface>
</bldg:boundedBy>
<bldg:interiorFurniture>
<bldg:BuildingFurniture>
<gml:name>Rocking Chair</gml:name>
<bldg:function codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/BuildingFurniture_fu
nction.xml"
>1340</bldg:function>
<bldg:lod4Geometry>
</gml:MultiSurface>
…
</gml:MultiSurface>
</bldg:lod4Geometry>
</bldg:BuildingFurniture>
</bldg:interiorFurniture>
</bldg:Room>
</bldg:interiorRoom>
<bldg:address>
…   (address see LOD1 example)
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
…   (see LOD1 example)
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

G.7. CityGML数据集示例：外观模型

以下CityGML数据集是基于G.2和G.3章中给出的LOD1和LOD2级别的简单建筑模型。此外，还定义了两个独立的外观主题——夏季主题和冬季主题——描述了建筑物和周围地形的不同外观。每个LOD精度级别的模型都拥有特定主题的独立外观。

此数据集中使用了CityGML外观模型的几个概念。在LOD1精度级别的模型中，一个*X3DMaterial*对象定义了整个建筑物的材质，同时该材质被应用于所有表面。此外，建筑物的地形和屋顶表面都被指定了含有地理参考的纹理材质（*GeoreferencedTexture*）。在LOD2中，建筑物的垂直表面单独使用参数化纹理材质（*ParameterizedTexture*），而屋顶表面和地形再次由含地理参考纹理材质（*GeoreferencedTexture*）表达。使用含地理参考纹理材质（*GeoreferencedTexture*）的对象使用ESRI世界文件（参见清单11）进行映射。以上建模方法得到了四种该数据集的可视化效果，如图82和图83所示。

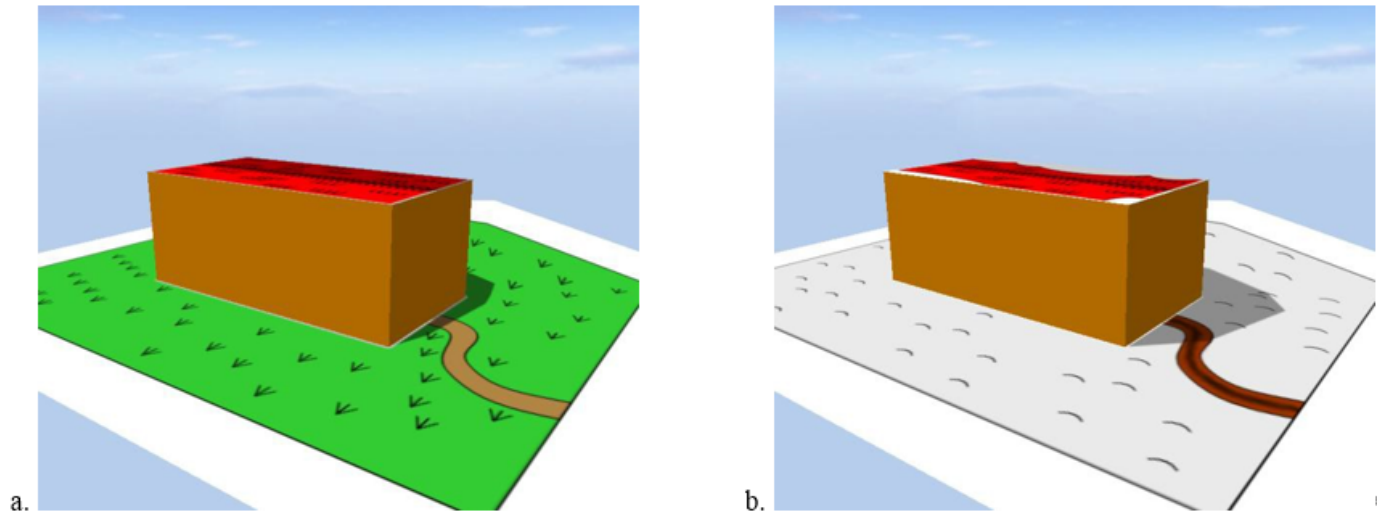

图 82. 使用CityGML的外观模型来可视化LOD1级别建筑模型。对于建筑模型和地形模型定义了两种显示主题：（a）左侧图表达了夏季显示风格，（b）右侧图片表达了冬季显示风格（图片来源：Hasso-Plattner-Institute）。

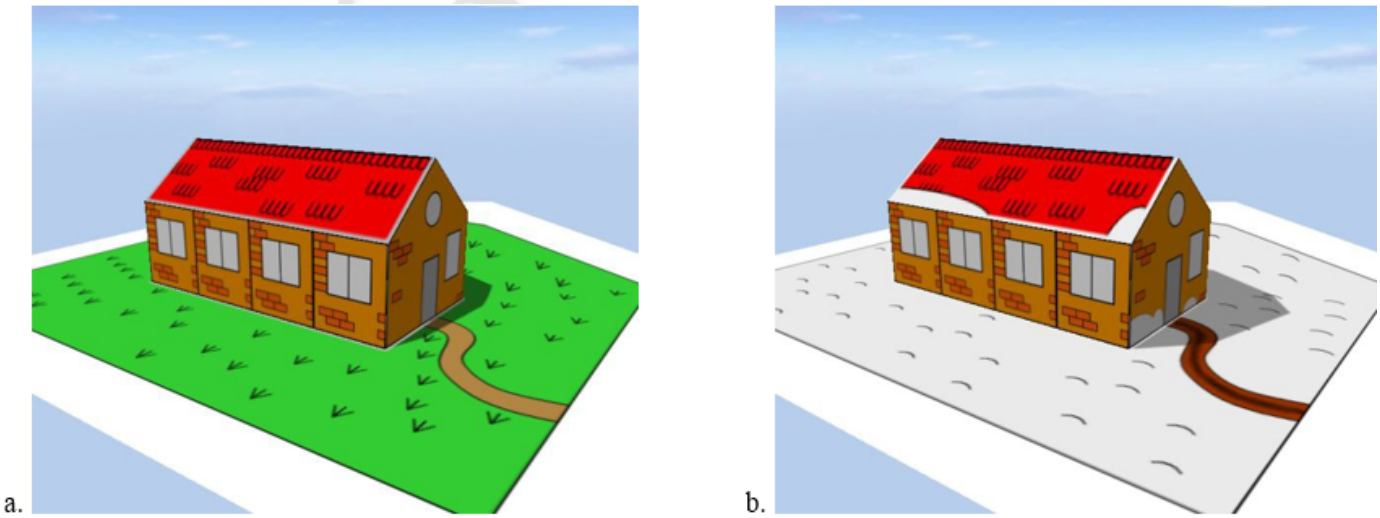

图 83. 使用CityGML的外观模型来可视化LOD2级别建筑模型。对于建筑模型和地形模型定义了两种显示主题：（a）左侧图表达了夏季显示风格，（b）右侧图片表达了冬季显示风格（图片来源：Hasso-Plattner-Institute）。

清单 9：来自CityGML数据集的摘录，说明了CityGML的外观模型。 数据集在图82和图83中可视化。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns="http://www.opengis.net/citygml/2.0" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/citygml/appearance/2.0" xmlns:dem=
"http://www.opengis.net/citygml/relief/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd
http://www.opengis.net/citygml/appearance/2.0
http://schemas.opengis.net/citygml/appearance/2.0/appearance.xsd">
<gml:boundedBy>
<gml:Envelope srsDimension="3" srsName=
"urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
<gml:lowerCorner>458868.0 5438343.0 112.0</gml:lowerCorner>
<gml:upperCorner>458892.0 5438362.0 117.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<bldg:function codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_fu
nction.xml">1000</bldg:function>
<bldg:yearOfConstruction>1985</bldg:yearOfConstruction>
<bldg:roofType codeSpace=
"http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_ro
ofType.xml">1030</bldg:roofType>
<bldg:measuredHeight uom="#m">5.0</bldg:measuredHeight>
<bldg:storeysAboveGround>1</bldg:storeysAboveGround>
<bldg:storeyHeightsAboveGround uom="#m">
3.0</bldg:storeyHeightsAboveGround>
<bldg:lod1Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface gml:id="lod1Surface">
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3">458875 5438350 112 458885 5438350 112 458885
5438350 116 458875
5438350 116 458875 5438350 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
```

```
</gml:surfaceMember>
…
<gml:surfaceMember>
<gml:Polygon gml:id="lod1RoofPoly1">
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3">458875 5438350 116 458885 5438350 116 458885
5438355 116 458875
5438355 116 458875 5438350 116 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
…
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod1Solid>
<bldg:lod2Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<gml:surfaceMember>
<gml:CompositeSurface gml:id="fLeft">
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing gml:id="fLeftExt1">
<gml:posList srsDimension="3">458875 5438350 112 458880 5438350 112 458880
5438350 115 458875
5438350 115 458875 5438350 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing gml:id="fLeftExt2">
<gml:posList srsDimension="3">458880 5438350 112 458885 5438350 112 458885
5438350 115 458880
5438350 115 458880 5438350 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:surfaceMember>
<gml:surfaceMember>
```

```
<gml:Polygon gml:id="fFront">
<gml:exterior>
<gml:LinearRing gml:id="fFrontExt">
<gml:posList srsDimension="3">458885 5438350 112 458885 5438355 112 458885
5438355 115 458885
5438352.5 117 458885 5438350 115 458885 5438350 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="fRight">
<gml:exterior>
<gml:LinearRing gml:id="fRightExt">
<gml:posList srsDimension="3">458885 5438355 112 458875 5438355 112 458875
5438355 115 458885
5438355 115 458885 5438355 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="fBack">
<gml:exterior>
<gml:LinearRing gml:id="fBackExt">
<gml:posList srsDimension="3">458875 5438355 112 458875 5438350 112 458875
5438350 115 458875
5438352.5 117 458875 5438355 115 458875 5438355 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="lod2RoofPoly1">
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3">458875 5438350 115 458885 5438350 115 458885
5438352.5 117 458875
5438352.5 117 458875 5438350 115 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="lod2RoofPoly2">
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3">458885 5438355 115 458875 5438355 115 458875
5438352.5 117 458885
```

```
5438352.5 117 458885 5438355 115 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3">458875 5438350 112 458875 5438355 112 458885
5438355 112 458885
5438350 112 458875 5438350 112 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod2Solid>
<bldg:address>
…   (address see LOD1 example)
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b ">
<dem:lod>1</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id=" GML_4eb161b0-aa7e-4087-937c-5c4c427c7fc9">
<gml:name>Ground</gml:name>
<dem:lod>1</dem:lod>
<dem:tin>
<gml:TriangulatedSurface gml:id="ground">
<gml:trianglePatches>
<gml:Triangle>
<gml:exterior>
<gml:LinearRing>
<gml:posList>458868 5438362 112 458875 5438355 112 458883 5438362 114
458868 5438362 112
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Triangle>
…
</gml:trianglePatches>
</gml:TriangulatedSurface>
</dem:tin>
</dem:TINRelief>
```

```
</dem:reliefComponent>
</dem:ReliefFeature>
</cityObjectMember>
<app:appearanceMember>
<app:Appearance>
<app:theme>Summer</app:theme>
<app:surfaceDataMember>
<app:X3DMaterial gml:id="lod1Material">
<app:diffuseColor>1.0 0.6 0.0</app:diffuseColor>
<app:target>#lod1Surface</app:target>
</app:X3DMaterial>
</app:surfaceDataMember>
<app:surfaceDataMember>
<app:GeoreferencedTexture>
<app:imageURI>ground_summer.png</app:imageURI>
<app:wrapMode>none</app:wrapMode>
<app:referencePoint>
<gml:Point>
<gml:pos srsDimension="2">458870 5438360</gml:pos>
</gml:Point>
</app:referencePoint>
<app:orientation>0.05 0.0 0.0 -0.05</app:orientation>
<app:target>#ground</app:target>
<app:target>#lod1RoofPoly1</app:target>
<app:target>#lod2RoofPoly1</app:target>
<app:target>#lod2RoofPoly2</app:target>
</app:GeoreferencedTexture>
</app:surfaceDataMember>
<app:surfaceDataMember>
<app:ParameterizedTexture gml:id="sideTexture">
<app:imageURI>facade.png</app:imageURI>
<app:wrapMode>wrap</app:wrapMode>
<app:target uri="#fLeft">
<app:TexCoordList>
<app:textureCoordinates ring="#fLeftExt1">0.0 0.0 2.0 0.0 2.0 1.0 0.0 1.0
0.0 0.0</app:textureCoordinates>
<app:textureCoordinates ring="#fLeftExt2">2.0 0.0 4.0 0.0 4.0 1.0 2.0 1.0
2.0 0.0</app:textureCoordinates>
</app:TexCoordList>
</app:target>
<app:target uri="#fRight">
<app:TexCoordGen>
<app:worldToTexture>-0.4 0.0 0.0 183550.0 0.0 0.0 0.3333 -37.3333 0.0 0.0
0.0 1.0</app:worldToTexture>
</app:TexCoordGen>
</app:target>
</app:ParameterizedTexture>
</app:surfaceDataMember>
<app:surfaceDataMember>
```

```
<app:ParameterizedTexture>
<app:imageURI>front_back_summer.png</app:imageURI>
<app:wrapMode>none</app:wrapMode>
<app:target uri="#fFront">
<app:TexCoordList gml:id="frontTexCoord">
<app:textureCoordinates ring="#fFrontExt">0.0 0.0 0.5 0.0 0.5 0.6 0.25 1.0
0.0 0.6 0.0 0.0</app:textureCoordinates>
</app:TexCoordList>
</app:target>
<app:target uri="#fBack">
<app:TexCoordList gml:id="backTexCoord">
<app:textureCoordinates ring="#fBackExt">0.5 0.0 1.0 0.0 1.0 0.6 0.75 1.0
0.5 0.6 0.5 0.0</app:textureCoordinates>
</app:TexCoordList>
</app:target>
</app:ParameterizedTexture>
</app:surfaceDataMember>
</app:Appearance>
</app:appearanceMember>
<app:appearanceMember>
<app:Appearance>
<app:theme>Winter</app:theme>
<app:surfaceDataMember>
<app:GeoreferencedTexture>
<app:imageURI>ground_winter.png</app:imageURI>
<app:wrapMode>none</app:wrapMode>
<app:referencePoint>
<gml:Point>
<gml:pos srsDimension="2">458870 5438360</gml:pos>
</gml:Point>
</app:referencePoint>
<app:orientation>0.05 0.0 0.0 -0.05</app:orientation>
<app:target>#ground</app:target>
<app:target>#lod1RoofPoly1</app:target>
<app:target>#lod2RoofPoly1</app:target>
<app:target>#lod2RoofPoly2</app:target>
</app:GeoreferencedTexture>
</app:surfaceDataMember>
<app:surfaceDataMember xlink:href="#lod1Material"/>
<app:surfaceDataMember xlink:href="#sideTexture"/>
<app:surfaceDataMember>
<app:ParameterizedTexture>
<app:imageURI>front_back_winter.png</app:imageURI>
<app:wrapMode>none</app:wrapMode>
<app:target uri="#fFront" xlink:href="#frontTexCoord"/>
<app:target uri="#fBack" xlink:href="#backTexCoord"/>
</app:ParameterizedTexture>
</app:surfaceDataMember>
</app:Appearance>
```

```
</app:appearanceMember>
</CityModel>
```

以下三张图片（图 84 - 图 86）应用于LOD2建筑立面所使用参数化纹理材质（*ParameterizedTexture*）。图像 *facade.png*（参见图 84）使用纹理环绕模式贴附于建筑立面，并分别应用于夏季和冬季外观主题。

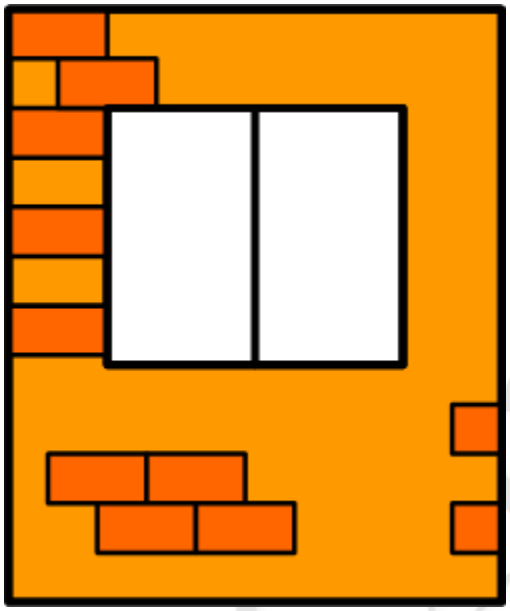

图 84. 图像facade.png是用于表达LOD2级别建筑立面材质的纹理贴图（参见图83.a.b）（图片来源：Hasso-Plattner-Institute）。

图85是LOD2精度级别模型在夏季外观主题下建筑正面和背面的纹理图集*front_back_summer.png*。该图像的部分区域做为立面纹理材质被分配给对应表面，此相关情况是通过*TextCoordList*对象来定义的。

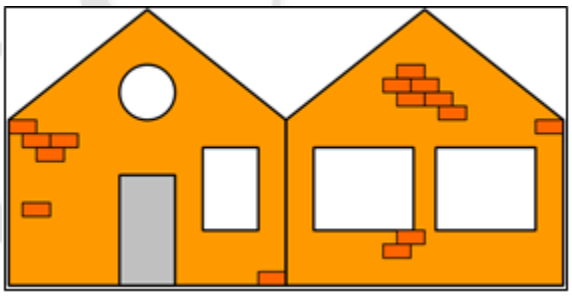

图 85. 图像front_back_summer.png是用于表达LOD2级别夏季外观主题建筑正面和背面材质的纹理图集（参见图83.a）（图片来源：Hasso-Plattner-Institute）。

与*front_back_summer.png*相同，纹理图集*front_back_winter.png*包含了冬季主题中 LOD2级别的建筑正面和北面的纹理。

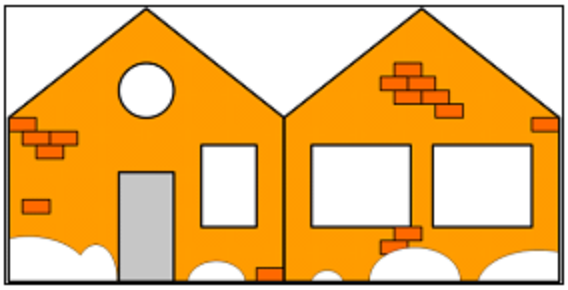

图 86. 图像front_back_winter.png是用于表达LOD2级别冬季外观主题建筑正面和背面材质的纹理图集（参见图83.b）（图片来源：Hasso-Plattner-Institute）。

图87和图 88中所示的光栅图像被分配给LOD1和LOD2中建筑物的地形和屋顶表面。在数据集中，这是由链接到相应GML几何对象*GeoreferencedTexture*实现的。图像*ground_summer.png*（参见图87）是表示夏季主题的纹理，而*ground_winter.png*（参见图88）用于冬季主题。

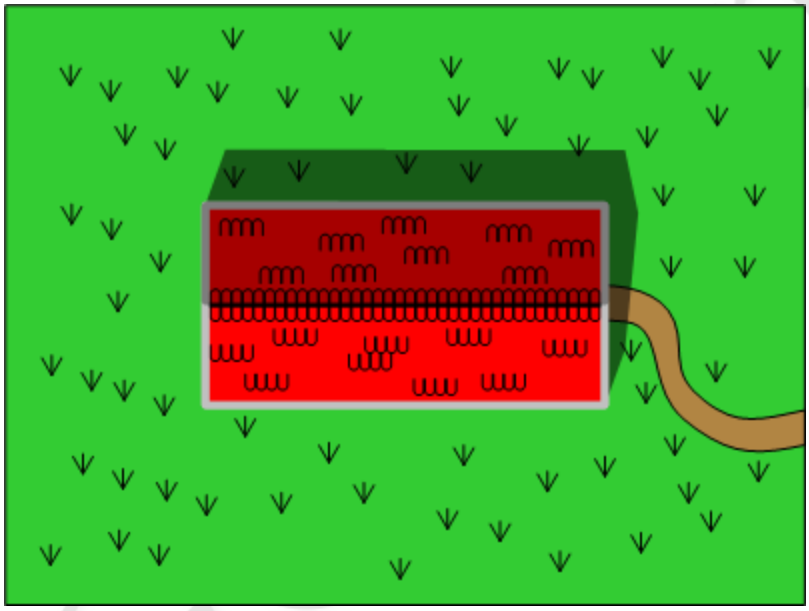

图 87. 图像ground_summer.png是用于夏季主题中LOD1和LOD2级别建筑物的地形和屋顶表面的纹理贴图（参见图 82 a和图 83 a）（图片来源：Hasso-Plattner-Institute）。

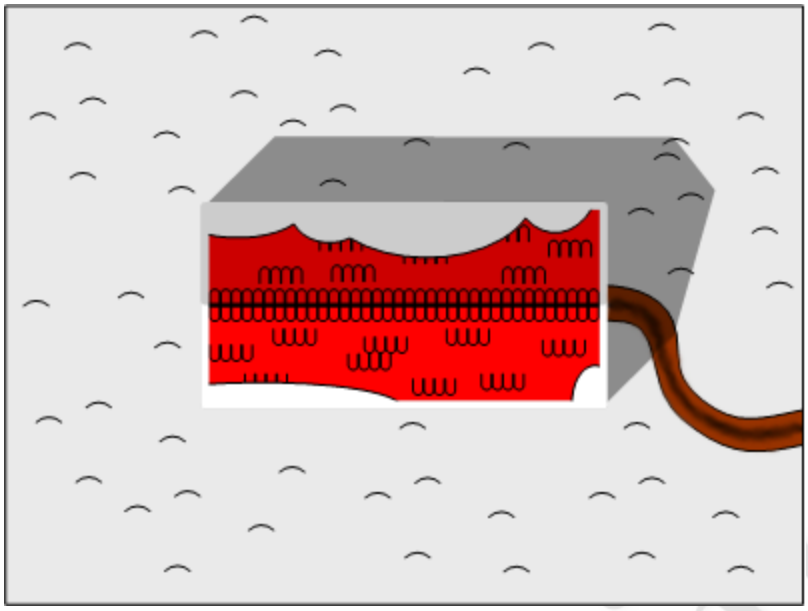

图 88. 图像ground_winter.png是用于冬季主题中LOD1和LOD2级别建筑物的地形和屋顶表面的纹理贴图（参见图 82 b和图 83 b）（图片来源：Hasso-Plattner-Institute）。

除了如清单9所示通过内部联结*GeoreferencedTexture*元素表示地理参考外，*Appearance*模块还支持使用带有图像文件的纹理映射（例如，使用地理参考TIFF）或提供单独的ESRI世界文件的方法。后者是一个六行文本文件，每行记录一个十进制数。这些数字对应于内联纹理映射的参数，并且可以进行1:1复制（反之亦然）。在ESRI世界文件的第1、3、2和 4 行中提供了由*GeoreferencedTexture*的*orientation*属性（由四个双精度浮点数组成的2x2行优先矩阵）所给出的旋转和缩放参数。*GeoreferencedTexture*对象中的*referencePoint*属性表示的世界空间左上角图像像素的中心位置对应于第5行和第6行。对于清单9中的示例，两个纹理图像的世界文件都包含以下值：

清单 10：世界文件的内容为 GeoreferencedTexture 指定相同的纹理映射，如清单 9 中所示的内联地理参考所给。

```
0.05
0.0
0.0
-0.05
458870
5438360
```

为了有效地使用这个世界文件，它必须与纹理图像一起存储（文件命名方式请参见 http://en.wikipedia.org/wiki/World_file ）。此外，必须将内联纹理映射从清单9中的两个*GeoreferencedTexture* 元素中删除，如以下摘录所示。用于表示*GeoreferencedTexture*的地理参考的两种替代方法都可在 http://schemas.opengis.net/citygml/examples/2.0/appearance/ 中找到。

清单 11：改编的GeoreferencedTexture元素，其纹理映射由清单10中所示的世界文件提供。

```
…
<app:GeoreferencedTexture>
<app:imageURI>ground_winter.png</app:imageURI>
<app:target>#ground</app:target>
<app:target>#lod1RoofPoly1</app:target>
<app:target>#lod2RoofPoly1</app:target>
<app:target>#lod2RoofPoly2</app:target>
</app:GeoreferencedTexture>
…
```

G.8. CityGML数据集示例：纹理坐标在含孔洞的复杂曲面中的应用

此案例介绍了纹理坐标在带孔洞的复杂表面的应用方法，并举例说明了覆盖的概念（参见第 9.1 章）。该案例表达了精度为LOD1 并且带有纹理的道路模型，此模型由一个环形交叉路口和两条相接的街道组成（图 89）。该模型包含两个使用了参数化纹理的对象（app:ParameterizedTexture）：一条是标准路段（rd，图 90），另一条是正在铺砌成标准道路的土路（dt，图 91）。所有几何图形都包含在gml:MultiSurface中。环形交叉路口由带孔洞的多边形 (roundaboutPoly)进行建模。交叉路口的外环 raEx 和内环raIn 都需要被赋予纹理坐标。
为了有效的贴图，需选择合适的贴图坐标和包裹模式，这样可在容许的纹理扭曲范围内对路段进行贴图。如对模型精度LOD的级别需求更高，则应使用无纹理失真的贴图方法，这种方法需要为环形交叉路口添加额外的路段纹理（图 92）或专门构建可用于完整环形交叉路口的纹理（图 93）。在这两种情况下，模型中使用到的有效纹理小于纹理原文件，从而造成了一定的纹理空间浪费，浪费的区域被标记为红色。

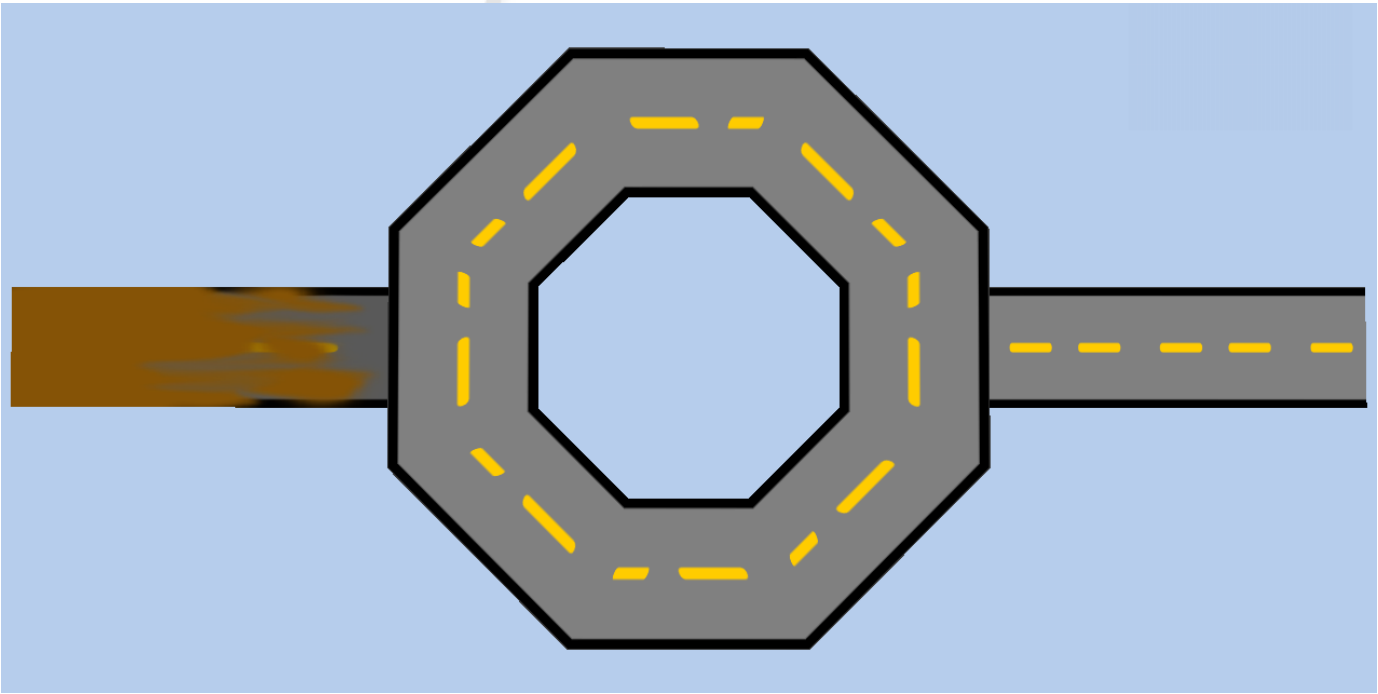

图 89. 带有材质的几何图像的渲染图（案例图片来源：Hasso-Plattner-Institute）。

图 90. 标准路段贴图纹理。

图 91. 土路路段贴图纹理。

图 92. 拐弯处路段的无失真贴图纹理。红色区域在模型中不会被显示，属于浪费的纹理。

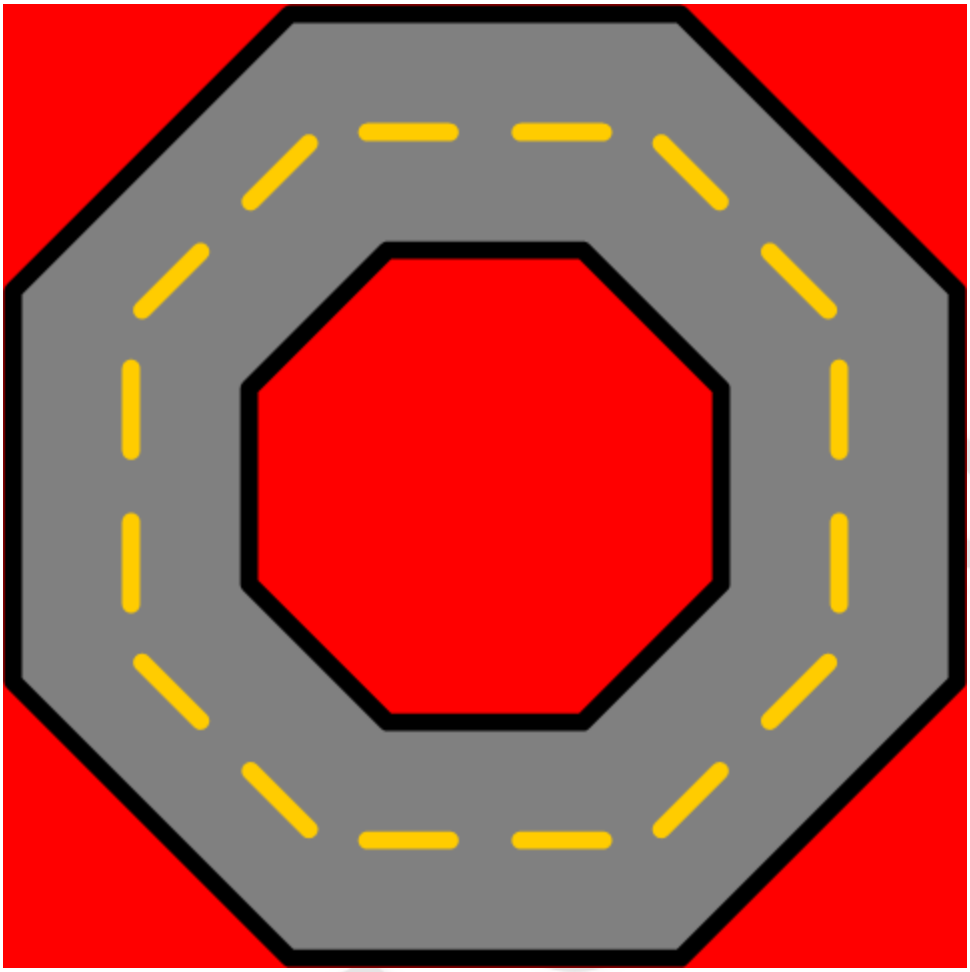

图 93. 完整环路无失真贴图纹理。红色区域在模型中不会被显示，属于浪费的纹理。

与环路相接的土路需要使用不同的纹理。即使它的几何体 (dirtPoly)包含于已经有纹理的道路中，现有的纹理也会被覆盖并替换。只有当需要给已有纹理的几何对象集合中的某个对象赋予新的纹理材质时，才会发生覆盖。对同一个表面几何对象直接赋予两个纹理是不允许的。这个规定同样适用于材质。

清单 12：CityGML数据集，说明了纹理坐标在带孔洞的复杂表面上的使用。 数据集在图 89 中可视化。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns="http://www.opengis.net/citygml/2.0" xmlns:app=
"http://www.opengis.net/citygml/appearance/2.0" xmlns:tran=
"http://www.opengis.net/citygml/transportation/2.0" xmlns:gml=
"http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/citygml/appearance/2.0
http://schemas.opengis.net/citygml/appearance/2.0/appearance.xsd
http://www.opengis.net/citygml/transportation/2.0
http://schemas.opengis.net/citygml/transportation/2.0/transportation.xsd">
<gml:boundedBy>
<!--The srsName attribute references a local engineering CRS that is
defined in the example dataset provided in annex G.9 -->
<gml:Envelope srsDimension="3" srsName="local-CRS-1">
<gml:lowerCorner>-45.0 -20.0 0.0</gml:lowerCorner>
```

```
<gml:upperCorner>45.0 20.0 10.0</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<tran:Road>
<app:appearance>
<app:Appearance>
<app:theme>visual</app:theme>
<app:surfaceDataMember>
<app:ParameterizedTexture gml:id="rd">
<app:imageURI>rd.png</app:imageURI>
<app:wrapMode>mirror</app:wrapMode>
<app:target uri="#road">
<app:TexCoordList>
<app:textureCoordinates ring="#raEx"> 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1
</app:textureCoordinates>
<app:textureCoordinates ring="#raIn"> 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
</app:textureCoordinates>
<app:textureCoordinates ring="#roadEx"> 0 0 2.5 0 2.5 1 0 1 0 0
</app:textureCoordinates>
<app:textureCoordinates ring="#dirtEx"> 0 0 2.5 0 2.5 1 0 1 0 0
</app:textureCoordinates>
</app:TexCoordList>
</app:target>
</app:ParameterizedTexture>
</app:surfaceDataMember>
<app:surfaceDataMember>
<app:ParameterizedTexture gml:id="dt">
<app:imageURI>dt.png</app:imageURI>
<app:wrapMode>mirror</app:wrapMode>
<app:target uri="#dirtPoly">
<app:TexCoordList>
<app:textureCoordinates ring="#dirtEx"> 0 0 1 0 1 1 0 1 0 0
</app:textureCoordinates>
</app:TexCoordList>
</app:target>
</app:ParameterizedTexture>
</app:surfaceDataMember>
</app:Appearance>
</app:appearance>
<tran:lod1MultiSurface>
<gml:MultiSurface gml:id="road">
<gml:surfaceMember>
<gml:Polygon gml:id="roundaboutPoly">

<gml:exterior>
<gml:LinearRing gml:id="raEx">
<gml:posList srsDimension="3"> -8 20 5 -20 8 5 -20 -8 5 -8 -20 5 8 -20 5
20 -8 5 20 8 5 8 20 5
```

```
-8 20 5 </gml:posList>
</gml:LinearRing>
</gml:exterior>
<gml:interior>
<gml:LinearRing gml:id="raIn">
<gml:posList srsDimension="3"> -4 10 5 4 10 5 10 4 5 10 -4 5 4 -10 5 -4
-10 5 -10 -4 5 -10 4 5
-4 10 5 </gml:posList>
</gml:LinearRing>
</gml:interior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="roadPoly">
<gml:exterior>
<gml:LinearRing gml:id="roadEx">
<gml:posList srsDimension="3"> 20 -4 5 45 -4 5 45 4 5 20 4 5 20 -4 5
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
<gml:Polygon gml:id="dirtPoly">
<gml:exterior>
<gml:LinearRing gml:id="dirtEx">
<gml:posList srsDimension="3"> -20 -4 5 -45 -4 5 -45 4 5 -20 4 5 -20 -4 5
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</tran:lod1MultiSurface>
</tran:Road>
</cityObjectMember>
</CityModel>
```

G.9. CityGML数据集示例：局部坐标参考系的应用

以下数据集演示了如何在CityGML中使用本地工程坐标系。数据集基于附录G.2中给出的LOD1示例。利用*CityModel*的*gml:metaDataProperty*对本地CRS进行定义，可用该功能三维坐标系进行描述。该原点对应于参考地球表面的锚定点（这里：在德国），该锚定点可在使用本地CRS的*EngineeringDatum*中进行定义。该CRS坐标系为组合的投影坐标系，其中ETRS89 / UTM zone 32N; EPSG code 25832坐标系用于平面测量，DHHN92 height, EPSG code 5783用于高度参考。*CityModel*的*gml:Envelope*，通过*gml:id “local-CRS-1”*引用本地CRS，因此*gml:Envelope*继承了*CityModel*所有属性中的几何体和特征。如果有需要，也可以使用*CityModel*的成员参照系对进行覆盖，例如，为每个城市对象提供单独的定位点。

清单 13：说明本地坐标参考系统使用的 CityGML 数据集。 数据集在第 272 页的图 76 中进行了可视化。

```
<?xml version="1.0" encoding="utf-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://www.opengis.net/citygml/2.0" xmlns:xAL=
"urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:dem="http://www.opengis.net/citygml/relief/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0"
xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0
http://schemas.opengis.net/citygml/building/2.0/building.xsd
http://www.opengis.net/citygml/relief/2.0
http://schemas.opengis.net/citygml/relief/2.0/relief.xsd ">
<gml:metaDataProperty>
<!-- Local EngineeringCRS definition contained specified inline as
metadata in this XML instance. -->
<!-- This CRS is referenced by geometry throughtout this instance by
srsName value #local-CRS-1 -->
<gml:EngineeringCRS xmlns:metadata="urn:x-ogp:spec:schema-
xsd:localmetadata" gml:id="local-CRS-1">
<gml:metaDataProperty>
<metadata:CommonMetaData>
<metadata:type>engineering</metadata:type>
</metadata:CommonMetaData>
</gml:metaDataProperty>
<gml:srsName codeSpace="XYZ">urn:ogc:def:crs:local:CRS:1</gml:srsName>
<gml:scope>CityGML</gml:scope>
<gml:usesCS>
<gml:CartesianCS gml:id="local-CS-1">
<gml:metaDataProperty>
<metadata:CommonMetaData>
<metadata:type>Cartesian</metadata:type>
<metadata:description>Cartesian 3D CS. Axes: UoM:
m.</metadata:description>
</metadata:CommonMetaData>
</gml:metaDataProperty>
<gml:csName codeSpace="XYZ">urn:ogc:def:crs:local:CS:1</gml:csName>
<gml:usesAxis>
<gml:CoordinateSystemAxis gml:id="local-axis-1" gml:uom=
"urn:ogc:def:uom:EPSG::9001">
<gml:name/>
<gml:axisID>
<gml:name>X</gml:name>
</gml:axisID>
<gml:axisAbbrev>x</gml:axisAbbrev>
<gml:axisDirection codeSpace="XYZ">X</gml:axisDirection>
</gml:CoordinateSystemAxis>
</gml:usesAxis>
<gml:usesAxis>
```

```
<gml:CoordinateSystemAxis gml:id="local-axis-2" gml:uom=
"urn:ogc:def:uom:EPSG::9001">
<gml:name/>
<gml:axisID>
<gml:name>Y</gml:name>
</gml:axisID>
<gml:axisAbbrev>y</gml:axisAbbrev>
<gml:axisDirection codeSpace="XYZ">Y</gml:axisDirection>
</gml:CoordinateSystemAxis>
</gml:usesAxis>
<gml:usesAxis>
<gml:CoordinateSystemAxis gml:id="local-axis-3" gml:uom=
"urn:ogc:def:uom:EPSG::9001">
<gml:name/>
<gml:axisID>
<gml:name>Z</gml:name>
</gml:axisID>
<gml:axisAbbrev>z</gml:axisAbbrev>
<gml:axisDirection codeSpace="XYZ">Z</gml:axisDirection>
</gml:CoordinateSystemAxis>
</gml:usesAxis>
</gml:CartesianCS>
</gml:usesCS>
<gml:usesEngineeringDatum>
<gml:EngineeringDatum gml:id="local-datum-1">
<gml:metaDataProperty>
<metadata:CommonMetaData>
<metadata:type>Cartesian datum</metadata:type>
</metadata:CommonMetaData>
</gml:metaDataProperty>
<gml:datumName codeSpace="XYZ">Datum1</gml:datumName>
<gml:anchorPoint codeSpace="
urn:ogc:def:crs,crs:EPSG::25832,crs:EPSG::5783">
458868.0 5438343.0 112.0</gml:anchorPoint>
<!-- The anchor point defines the origin of the local CS with respect to
the world CRS -->
<!-- In this example, the anchor point references a point on the earth (in
Germany) using a compound CRS -->
<!-- For planimetry, the reference system ETRS89 / UTM zone 32N (EPSG code
25832) is used -->
<!-- The vertical reference system is DHHN92 height (EPSG code 5783) -->
</gml:EngineeringDatum>
</gml:usesEngineeringDatum>
</gml:EngineeringCRS>
</gml:metaDataProperty>
<gml:description>Simple example for a CityGML dataset using a local
engineering CRS</gml:description>
<gml:name>Simple 3D city model LOD1 without Appearance</gml:name>
<gml:boundedBy>
```

```
<gml:Envelope srsName="#local-CRS-1">
<!-- Encoding of local-CRS-1 is specified in CityModel metadataProperty in
this document-->
<gml:pos srsDimension="3">0.0 0.0 0.0</gml:pos>
<gml:pos srsDimension="3">24.0 19.0 4.0</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
<gml:name>Example Building LOD1 </gml:name>
…   (Attributes see example LOD1)
<bldg:lod1Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface gml:id="lod1Surface">
<!-- Face Side 1 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>7.0 7.0 0.0 17.0 7.0 0.0 17.0 7.0 4.0 7.0 7.0 4.0 7.0 7.0
0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 2 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>17.0 7.0 0.0 17.0 12.0 0.0 17.0 12.0 4.0 17.0 7.0 4.0 17.0
7.0 0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 3 -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>17.0 12.0 0.0 7.0 12.0 0.0 7.0 12.0 4.0 17.0 12.0 4.0 17.0
12.0 0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Side 4 -->
```

```
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>7.0 12.0 0.0 7.0 7.0 0.0 7.0 7.0 4.0 7.0 12.0 4.0 7.0 12.0
0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Top -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>7.0 7.0 4.0 17.0 7.0 4.0 17.0 12.0 4.0 7.0 12.0 4.0 7.0 7.0
4.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
<!-- Face Bottom -->
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList>7.0 7.0 0.0 7.0 12.0 0.0 17.0 12.0 0.0 17.0 7.0 0.0 7.0 7.0
0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod1Solid>
<bldg:address>
<Address>
…
</Address>
</bldg:address>
</bldg:Building>
</cityObjectMember>
<cityObjectMember>
<dem:ReliefFeature gml:id="GML_6bb30328-7599-4500-90ef-766fde6aa67b">
<gml:name>Example TIN LOD1</gml:name>
<dem:lod>1</dem:lod>
<dem:reliefComponent>
<dem:TINRelief gml:id="GUID_04D4DsNGv1MfvYu5O3lkcW">
```

```
<gml:name>Ground</gml:name>
<dem:lod>1</dem:lod>
<dem:tin>
<gml:TriangulatedSurface gml:id="ground">
<gml:trianglePatches>
<gml:Triangle>
<gml:exterior>
<gml:LinearRing>
<gml:posList>0.0 19.0 0.0 7.0 12.0 0.0 15.0 19.0 2.0 0.0 19.0
0.0</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Triangle>
<gml:Triangle>
…

</gml:Triangle>
… (more triangles)
</gml:trianglePatches>
</gml:TriangulatedSurface>
</dem:tin>
</dem:TINRelief>
</dem:reliefComponent>
</dem:ReliefFeature>
</cityObjectMember>
</CityModel>
```

Annex H: （资料性附录）ADE在噪声干扰模拟中的应用示例

本附件介绍了如何在环境模拟应用程序中使用CityGML。相应的应用领域扩展 (ADE) 的定义包含在示例中。

欧盟环境噪声指令（2002/49/EG）要求欧盟成员国每5年计算一次建筑物4m高度处的噪声，并将结果记录在噪声图中。噪声图可为欧盟和受噪声影响的公民提供相关信息（图 94）。这些噪声图是基于声学模型和噪声传播计算得到的，并不是实际测量结果（图 95）。对于噪声传播计算，每个欧盟成员国都需要大量的专题数据和三维地理数据。由于需要进行噪声计算的空间范围很大，且大部分用户对数据颗粒度要求较高，因此为众多用户提供全域建筑物、道路、铁路和地形的三维地理数据是非常必要的。

道路噪音水平的计算需要交通流量、重型车辆占比、道路速度限制、路面类型和道路坡度等相关信息。此外，噪音水平取决于发射点和接收点之间的距离以及反射（例如在建筑物立面上）或屏蔽效果（例如噪声屏障）。噪音水平分别计算白天 (06.00-18.00)、晚上 (18.00-22.00) 和夜间 (22.00-06.00)三个时间段。由于噪音水平是在4m高度计算的，并且考虑到垂直反射面的影响（例如隔音屏障和建筑物），因此需要大量的三维地理数据。除了所有 三维地理数据之外，还需要特定的专题数据。例如，道路噪声计算还需要以下数据：10m网格的数字地形模型、具有主题属性（例如反射、居民）的三维建筑模型、具有主题属性（例如交通流量、重型车辆百分比）的三维道路数据、速度限制、路面类型、道路坡度、道路宽度）、三维隔音屏障及其主题属性（例如反射）。

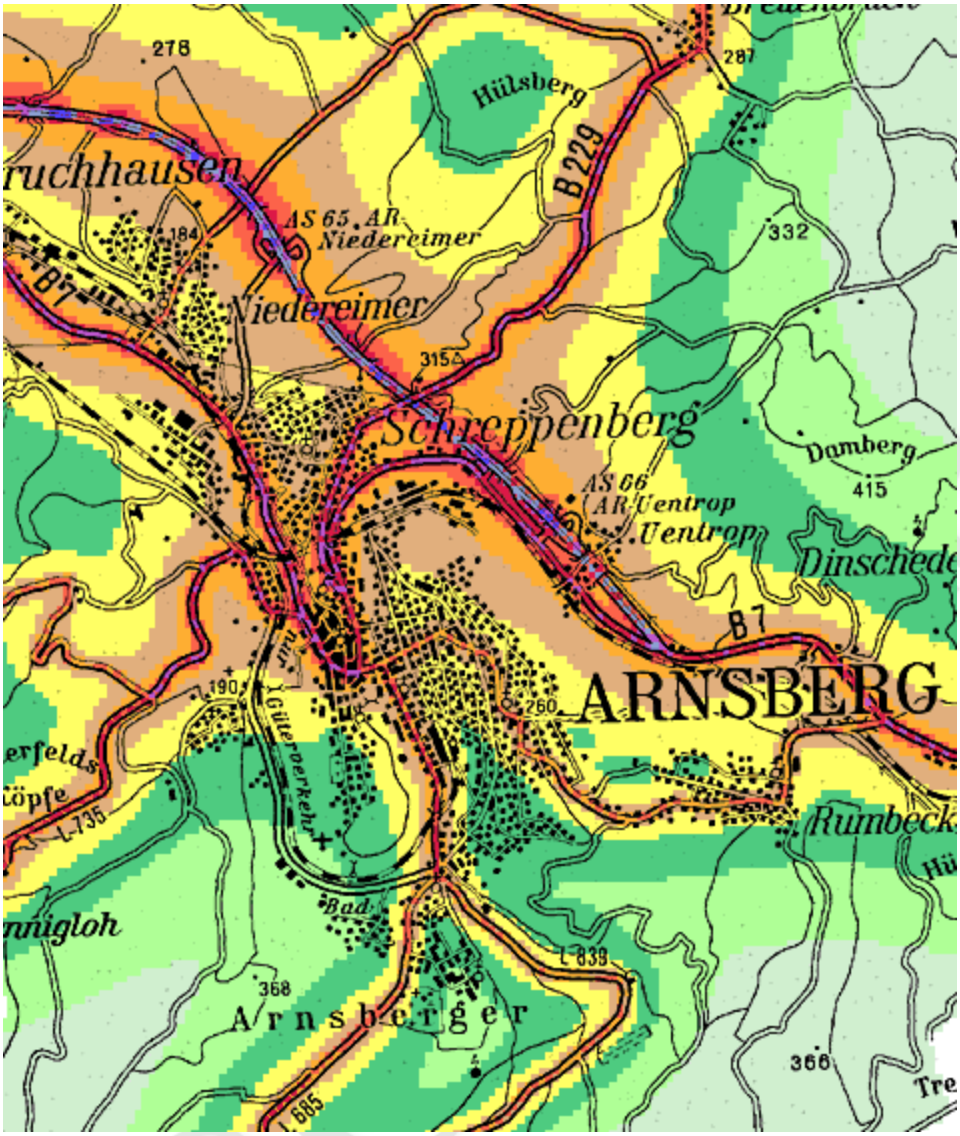

图 94. 图像ground_winter.png是用于冬季主题中LOD1和LOD2精度级别建筑物的地形和屋顶表面的纹理贴图（参见图 82 b和图 83 b）（图片来源：Hasso-Plattner-Institute）。

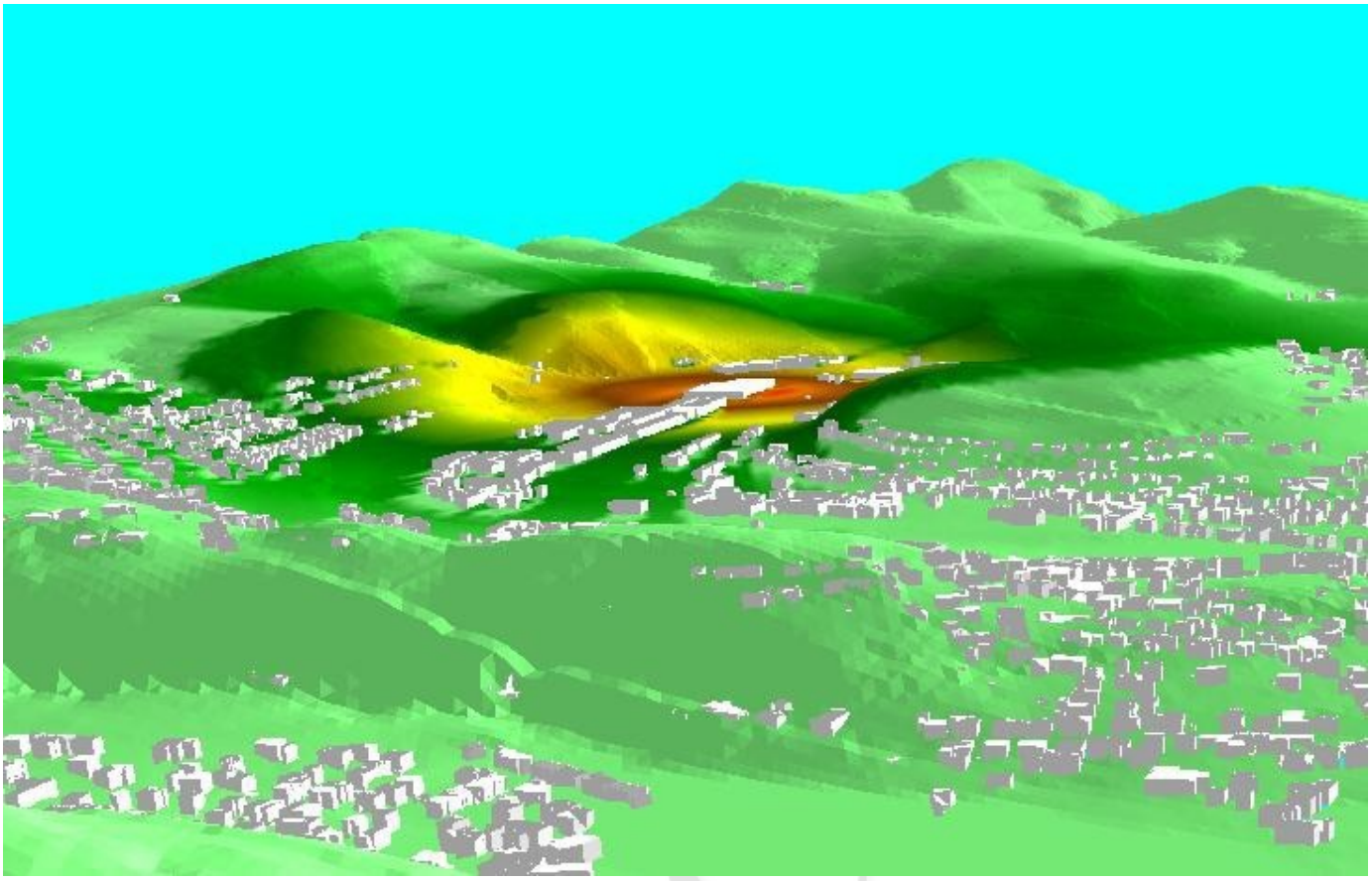

图 95. 在生成图 94 中的噪声图之前，第一步需使用噪声计算软件基于三维CityGML地理数据对噪声发射源进行建模（来源：Surveying and Mapping Agency NRW, Stapelfeldt GmbH, Institute of Geodesy and Geoinfor-mation Uni Bonn）。

在北莱茵-威斯特法伦州，必须考虑一种特殊情况：由于该区域人口和交通路线密度高，因此是德国噪声计算区域最大和对象数量最多的地区。该项目的目的是为5年迭代周期和不同的噪声计算机构提供可持续、高效、多源的三维地理数据。

为了提供如此大量的全域三维地理数据，该项目的责任方，特别是北莱茵-威斯特法伦州环境、自然保护、农业和消费者保护部、北莱茵州自然、环境和消费者保护局-威斯特法伦州和北莱茵-威斯特法伦州测绘局决定使用北莱茵-威斯特法伦州的空间数据基础设施 (GDI NRW)，并将其扩展为全域范围的2.5D和3D地理信息网络服务。因此，实施了用于建筑模型、地形、道路和铁路数据的新OGC Web服务（例如，用于LOD1中的三维体块模型、三维道路和铁路数据的网页特征服务，以及用于DTM的网页覆盖服务）。

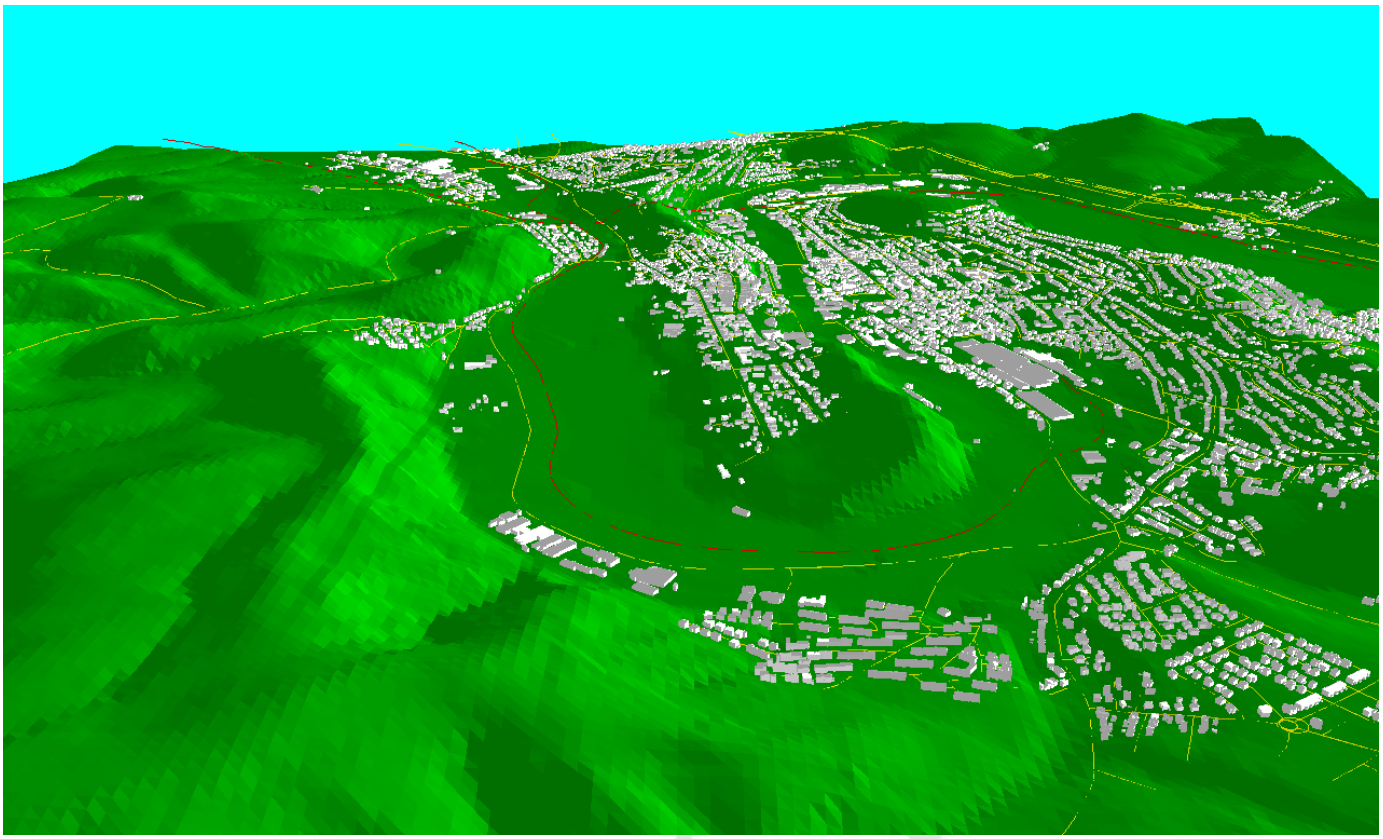

图 96. CityGML中的三维地理数据，用于计算图94中的噪声图：GeoTiff 中的DTM、CityGML 中的三维体块模型、CityGML 中的三维道路和铁路数据、CityGML 中更高级别道路的州道数据（来源：Surveying and Mapping Agency NRW, State Road Enterprice NRW, Stapelfeldt GmbH, Institute of Geodesy and Geoinformation Uni Bonn）。

CityGML与GeoTIFF一起用作网络服务和噪声计算软件之间的唯一交换格式（图 96 -图 98）。针对噪声指令的特殊要求，波恩大地测量和地理信息大学研究所和GDI NRW的特别兴趣小组SIG 3D共同开发了CityGML噪声应用模式。它基于 ADE 机制（参见第6.12和10.13章）。这种机制允许通过主题属性补充CityGML中现有的类和对象（例如建筑物）。这些属性的数量和类型是可选择的。CityGML模式还可以通过新类进行补充。因此，噪声应用模式包含新对象（例如根据噪声要求分割道路 - *NoiseRoadSegment*，图 99）以及附加到现有对象的噪声属性（例如建筑物的反射，图 100）。这些额外的噪声属性源自德国联邦政府为履行欧盟环境噪声指令而颁布的法规（参见 BImSchV 2006、VBUS 2006、VBUSch 2006）。

该项目的交互技术展示了一项重要的创新，因为这是第一次通过通用标准和网络服务提供全域范围的三维地理数据（参见 Czerwinski 等人，2006b）。因此，北莱茵-威斯特法伦州用于噪声计算的空间数据基础设施为欧盟2007/2/EC欧洲空间信息基础设施）的 INSPIRE指令提供了一个应用示例（参见 Czerwinski 等人，2007 年）。

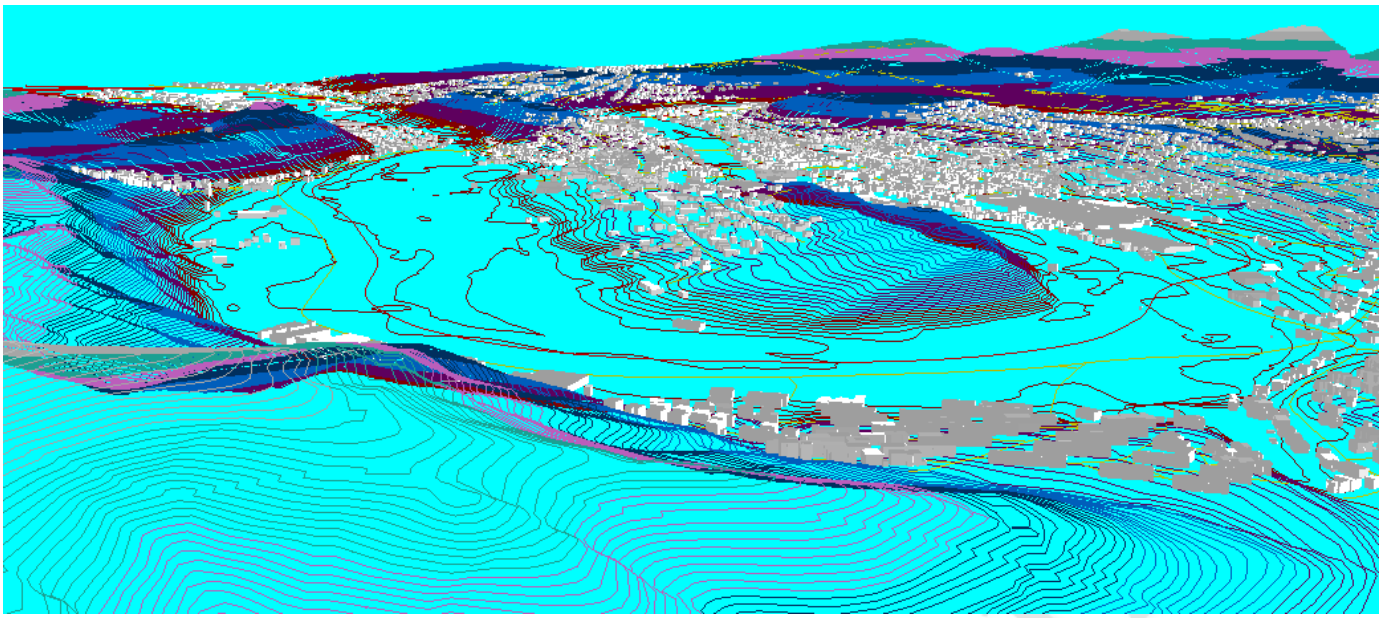

图 97. CityGML中的三维地理数据，用于计算图94中的噪声图：用于生成CityGML隔断线的等高线、CityGML 中的三维体块模型、CityGML 中的三维公路和铁路数据和CityGML中更高级别道路的数据（来源：Surveying and Mapping Agency NRW, State Road Enterprise NRW, Stapelfeldt GmbH, Institute of Geodesy and Geoinformation Uni Bonn)。

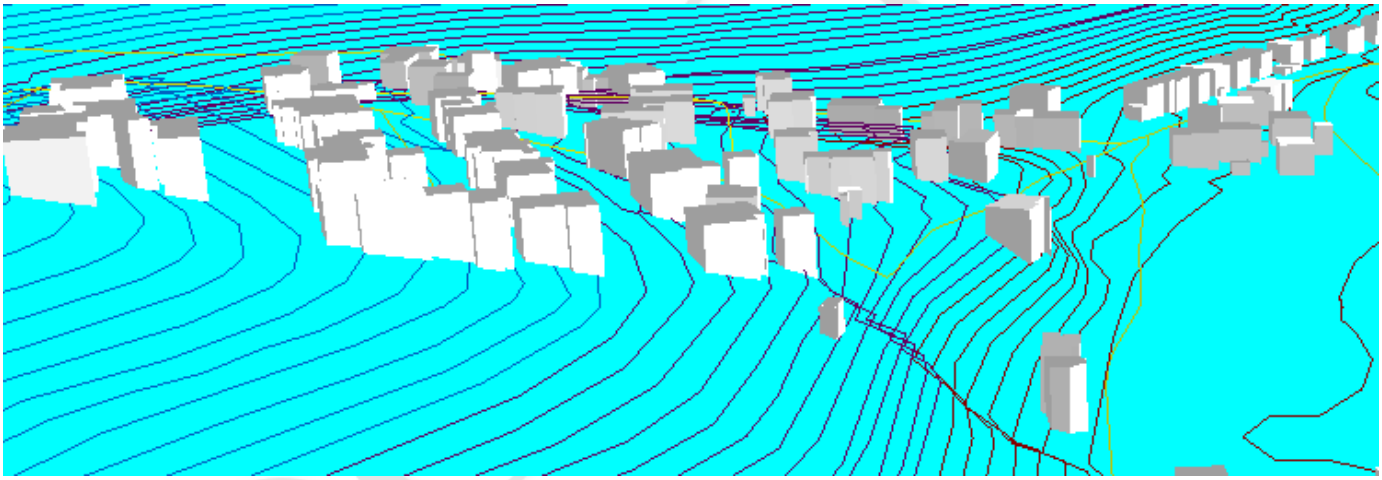

图 98. 此图为截取图97局部后放大的图，表明如何通过适当的CityGML建模将三维体块模型集成到DTM中（以ALK建筑物多边形的最低点作为生成建筑物底部的基准点）（来源：Surveying and Mapping Agency NRW, Stapelfeldt GmbH, Institute of Geodesy and Geoinformation Uni Bonn)。

H.1. CityGML噪声应用领域扩展

在本节中，CityGML噪声应用领域扩展的数据模型以UML图和XML模式给出。由于特定属性和对象类型的语义来源于德国关于噪声模拟计算的规定，因此这里不对其进行详细解释（参见BimSchV 2006、VBUS 2006、VBUSch 2006）。本节的目的是提供一个使用应用领域扩展机制扩展CityGML的示例。CityGML噪声应用领域扩展的XML模式定义，及示例数据集，可以从 http://schemas.opengis.net/citygml/examples/2.0/ade/noise-ade/ 获得。

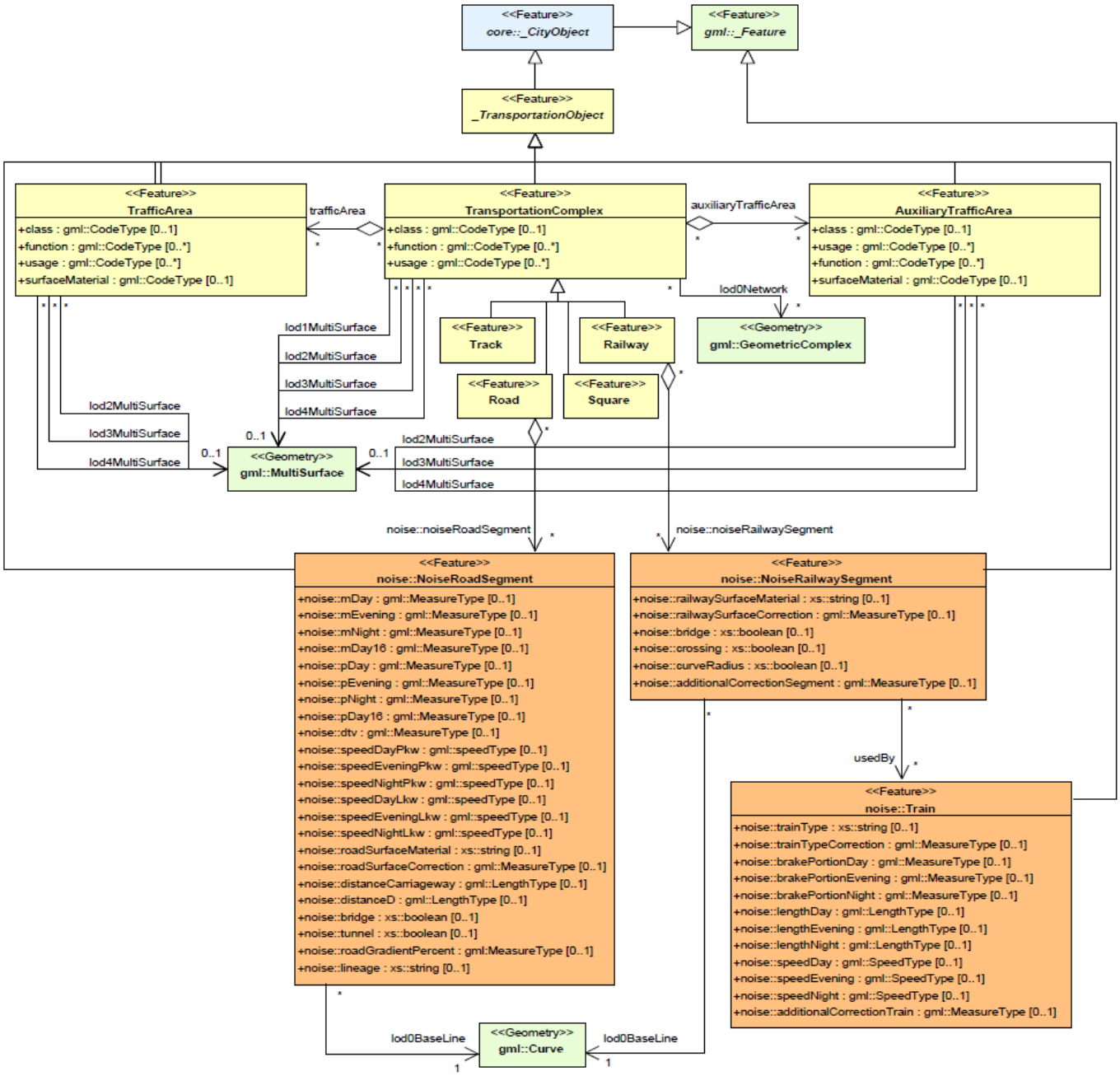

图 99. CityGML噪声应用模式——交通模型（浅黄色=CityGML交通模块，浅橙色=CityGML噪声应用程序扩展）。前缀被用于指示与模型元素相关联的XML命名空间。没有前缀的元素名称在CityGML交通模块中定义。CityGML噪声应用程序扩展带有前缀“noise”（来源：波恩大学大地测量和地理信息研究所）。

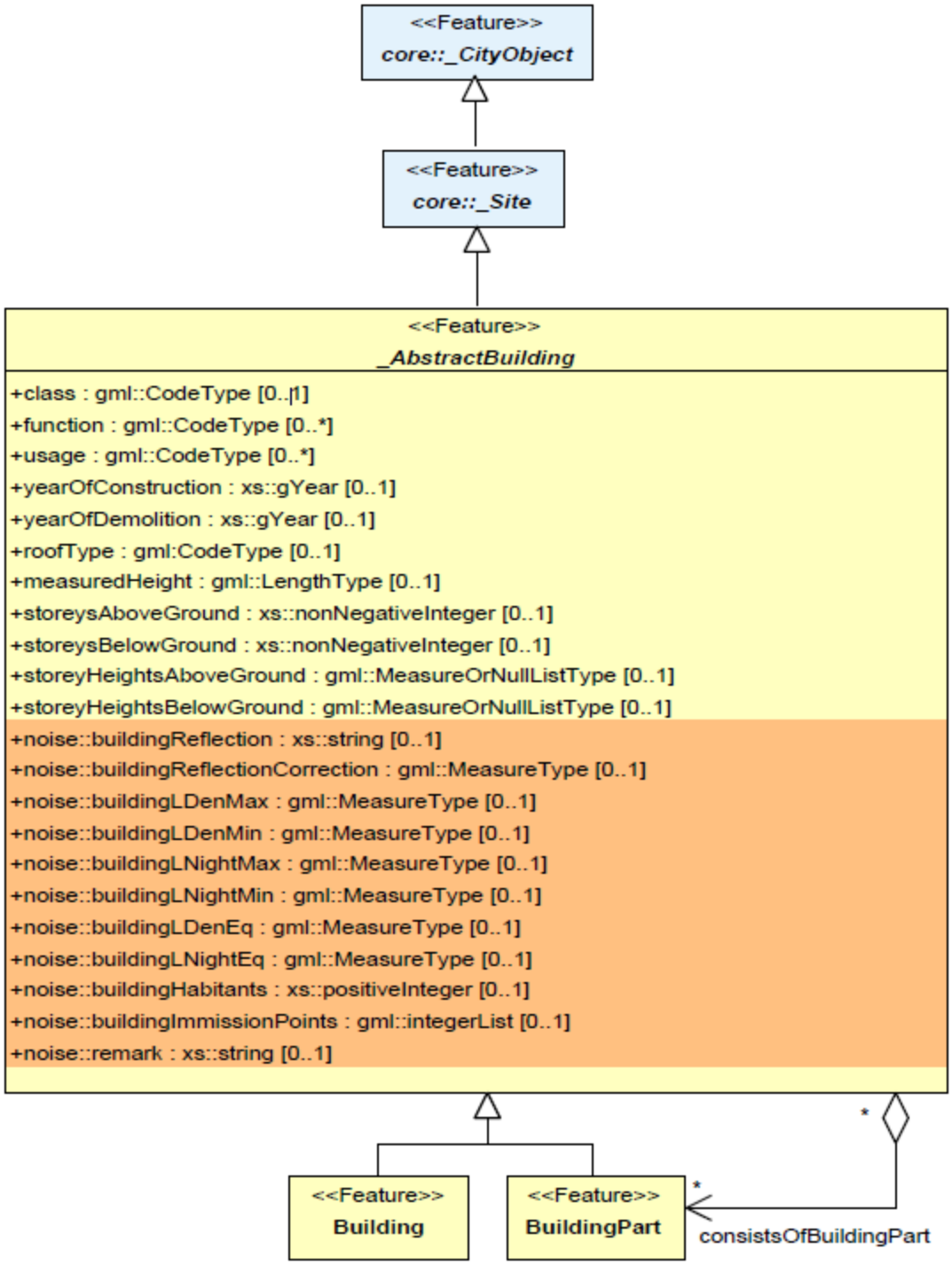

图 100. CityGML噪声应用模式——建筑模型的摘录（浅黄色=CityGML建筑模块，浅橙色=CityGML噪声应用程序扩展)。前缀用于指示与模型元素关联的XML命名空间。不带前缀的元素名称在CityGML建筑模块中定义。CityGML噪声应用程序扩展带有前缀“noise”（来源：波恩大学大地测量和地理信息研究所）。

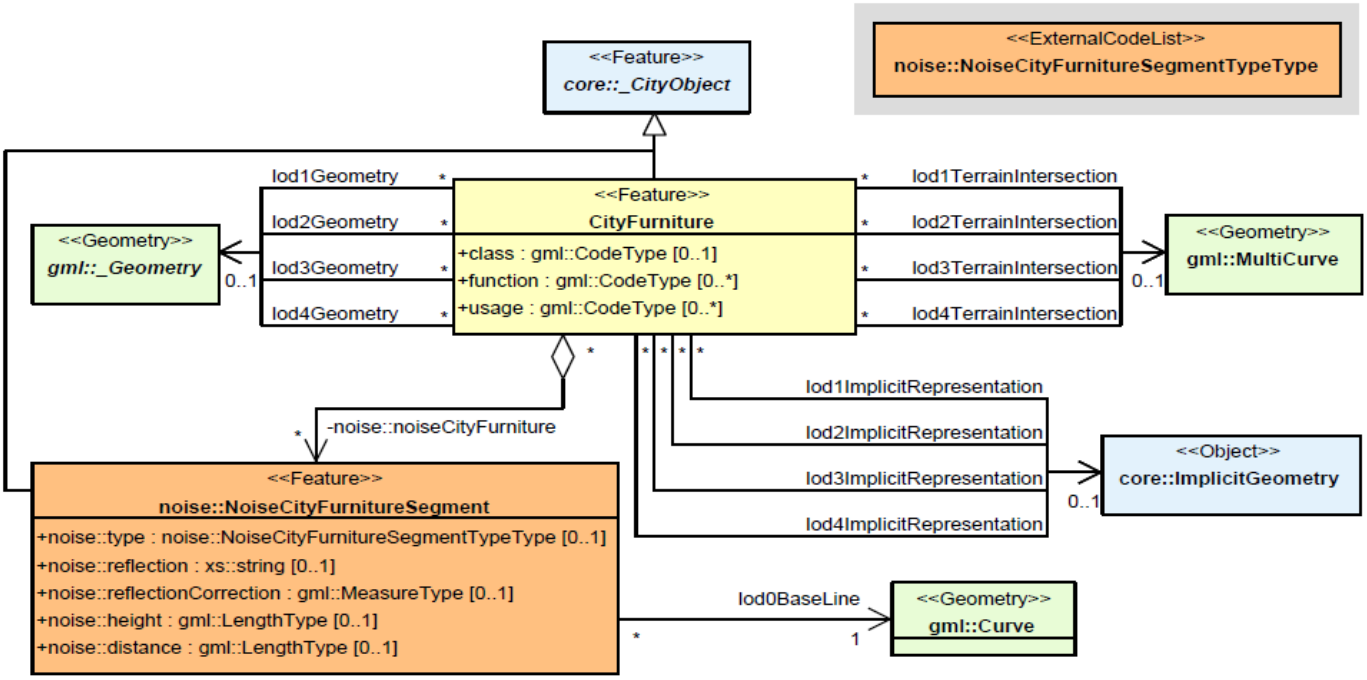

图 101. CityGML噪声应用模式——城市家具模型（浅黄色=CityGML城市家具模块，浅橙色=CityGML噪声应用程序扩展)。前缀用于指示与模型元素关联的XML命名空间。没有前缀的元素名称在CityGML的城市家具模块中定义。CityGML噪声应用程序扩展带有前缀“noise”（来源：波恩大学大地测量和地理信息研究所）。

噪声应用程序扩展模式定义文件的表头

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.citygml.org/ade/noise_de/2.0" xmlns:gml=
"http://www.opengis.net/gml" xmlns:core=
"http://www.opengis.net/citygml/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0" xmlns:frn=
"http://www.opengis.net/citygml/cityfurniture/2.0" xmlns:tran=
"http://www.opengis.net/citygml/transportation/2.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"targetNamespace="http://www.ci
tygml.org/ade/noise_de/2.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xsd:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xsd:import namespace="http://www.opengis.net/citygml/2.0" schemaLocation
="http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"/>
<xsd:import namespace="http://www.opengis.net/citygml/transportation/2.0"
schemaLocation="http://schemas.opengis.net/citygml/transportation/2.0/tran
sportation.xsd"/>
<xsd:import namespace="http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://schemas.opengis.net/citygml/building/2.0/building.x
sd"/>
<xsd:import namespace="http://www.opengis.net/citygml/cityfurniture/2.0"
schemaLocation="http://schemas.opengis.net/citygml/cityfurniture/2.0/cityF
urniture.xsd"/>
…
</xsd:schema>
```

NoiseCityFurnitureSegmentTypeType,NoiseCityFurnitureSegment

```
<xsd:element name="noiseCityFurnitureSegmentProperty" type=
"NoiseCityFurnitureSegmentPropertyType" substitutionGroup=
"frn:_GenericApplicationPropertyOfCityFurniture"/>
<!--
========================================================================
================= -->
<xsd:complexType name="NoiseCityFurnitureSegmentPropertyType">
<xsd:sequence minOccurs="0">
<xsd:element ref="NoiseCityFurnitureSegment" minOccurs="0"/>
</xsd:sequence>
<xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xsd:complexType>
<!--
========================================================================
================= -->
<xsd:complexType name="NoiseCityFurnitureSegmentType">
<xsd:complexContent>
<xsd:extension base="core:AbstractCityObjectType">
<xsd:sequence>
<xsd:element name="type" type="gml:CodeType" minOccurs="0"/>
<xsd:element name="reflection" type="xsd:string" minOccurs="0"/>
<xsd:element name="reflectionCorrection" type="gml:MeasureType" minOccurs
="0"/>
<xsd:element name="height" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="distance" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="lod0BaseLine" type="gml:CurvePropertyType"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!--
========================================================================
================= -->
<xsd:element name="NoiseCityFurnitureSegment" type=
"NoiseCityFurnitureSegmentType" substitutionGroup="core:_CityObject"/>
```

NoiseRoadSegmentType, NoiseRoadSegment

```
<xsd:element name="noiseRoadSegmentProperty" type=
"NoiseRoadSegmentPropertyType" substitutionGroup=
"tran:_GenericApplicationPropertyOfRoad"/>
<!--
========================================================================
================= -->
<xsd:complexType name="NoiseRoadSegmentPropertyType">
<xsd:sequence minOccurs="0">
<xsd:element ref="NoiseRoadSegment"/>
```

```
</xsd:sequence>
<xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xsd:complexType>
<!--
=============================================================================
================= -->
<xsd:complexType name="NoiseRoadSegmentType">
<xsd:complexContent>
<xsd:extension base="tran:AbstractTransportationObjectType">
<xsd:sequence>
<xsd:element name="mDay" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="mEvening" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="mNight" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="mDay16" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="pDay" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="pEvening" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="pNight" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="pDay16" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="dtv" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="speedDayPkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedEveningPkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedNightPkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedDayLkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedEveningLkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedNightLkw" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="roadSurfaceMaterial" type="xsd:string" minOccurs="0"/>
<xsd:element name="roadSurfaceCorrection" type="gml:MeasureType"
minOccurs="0"/>
<xsd:element name="distanceCarriageway" type="gml:LengthType" minOccurs="
0"/>
<xsd:element name="distanceD" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="bridge" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="tunnel" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="roadGradientPercent" type="gml:MeasureType" minOccurs=
"0"/>
<xsd:element name="lod0BaseLine" type="gml:CurvePropertyType"/>
<xsd:element name="lineage" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!--
=============================================================================
================= -->
<xsd:element name="NoiseRoadSegment" type="NoiseRoadSegmentType"
substitutionGroup="core:_CityObject"/>
```

NoiseRailwaySegmentType,NoiseRailwaySegment

```
<xsd:complexType name="NoiseRailwaySegmentPropertyType">
<xsd:sequence minOccurs="0">
<xsd:element ref="NoiseRailwaySegment"/>
</xsd:sequence>
<xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xsd:complexType>
<!--
================================================================================
================== -->
<xsd:complexType name="NoiseRailwaySegmentType">
<xsd:complexContent>
<xsd:extension base="tran:AbstractTransportationObjectType">
<xsd:sequence>
<xsd:element name="railwaySurfaceMaterial" type="xsd:string" minOccurs="0
"/>
<xsd:element name="railwaySurfaceCorrection" type="gml:MeasureType"
minOccurs="0"/>
<xsd:element name="bridge" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="crossing" type="xsd:boolean" minOccurs="0"/>
<xsd:element name="curveRadius" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="additionalCorrectionSegment" type="gml:MeasureType"
minOccurs="0"/>
<xsd:element name="lod0BaseLine" type="gml:CurvePropertyType"/>
<xsd:element name="usedBy" type="TrainPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!--
================================================================================
================== -->
<xsd:element name="NoiseRailwaySegment" type="NoiseRailwaySegmentType"
substitutionGroup="core:_CityObject"/>
```

TrainType, TrainPropertyType

```
<xsd:complexType name="TrainPropertyType">
<xsd:sequence>
<xsd:element name="Train" type="TrainType"/>
</xsd:sequence>
<xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xsd:complexType>
<!--
================================================================================
================= -->
<xsd:complexType name="TrainType">
<xsd:complexContent>
<xsd:extension base="gml:AbstractFeatureType">
<xsd:sequence>
<xsd:element name="trainType" type="xsd:string"/>
<xsd:element name="trainTypeCorrection" type="gml:MeasureType" minOccurs=
"0"/>
<xsd:element name="brakePortionDay" type="gml:MeasureType" minOccurs="0"/>
<xsd:element name="brakePortionEvening" type="gml:MeasureType" minOccurs=
"0"/>
<xsd:element name="brakePortionNight" type="gml:MeasureType" minOccurs="0
"/>
<xsd:element name="lengthDay" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="lengthEvening" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="lengthNight" type="gml:LengthType" minOccurs="0"/>
<xsd:element name="speedDay" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedEvening" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="speedNight" type="gml:SpeedType" minOccurs="0"/>
<xsd:element name="additionalCorrectionTrain" type="gml:MeasureType"
minOccurs="0"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

Application specific attributes for _AbstractBuilding

```
<xsd:element name="buildingReflection" type="xsd:string"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingReflectionCorrection" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLDenMax" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLDenMin" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLDenEq" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLNightMax" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLNightMin" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingLNightEq" type="gml:MeasureType"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingHabitants" type="xsd:positiveInteger"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingAppartments" type="xsd:positiveInteger"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="buildingImmissionPoints" type="gml:integerList"
substitutionGroup="bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
<xsd:element name="remark" type="xsd:string" substitutionGroup=
"bldg:_GenericApplicationPropertyOfAbstractBuilding"/>
```

H.2. 示例数据集

以下数据集阐述了一份使用应用程序噪声模式的CityGML实例文档。它包含两个城市对象（CityObject）要素：道路对象和建筑对象。该数据集引用了CityGML噪声应用程序扩展的XML模式定义文件。该文件特地导入了部分CityGML模块的XML模式定义。这些CityGML模块（包括*CityGML Core*,*Building*,*Transportation*和*CityFurniture*）由噪声应用程序扩展所扩展而来。因此，被采用的CityGML模块所定义的所有类都可以在实例文档中使用。此外，还可以使用应用程序特定的添加项，例如新对象类型（例如*NoiseRoadSegment*），和附加的主题属性（例如为*_AbstractBuilding*定义的属性）。这些附加元素与标准CityGML元素的不同之处在于，前者的命名空间的前缀*noise*是指噪声的模式定义。

列表14：实现所示CityGML噪声应用模式的CityGML数据集

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CityModel xmlns="http://www.opengis.net/citygml/2.0" xmlns:tran=
"http://www.opengis.net/citygml/transportation/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0" xmlns:noise=
"http://www.citygml.org/ade/noise_de/2.0" xmlns:gml=
"http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xAL="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.citygml.org/ade/noise_de NoiseADE/CityGML-NoiseADE.xsd">
<gml:boundedBy>
```

```
<gml:Envelope srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
<gml:pos srsDimension="3">5616000.0 2540097.5 54.5</gml:pos>
<gml:pos srsDimension="3">5673522.3 2576495.6 172.9</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<tran:Road gml:id="CR_0815">
<gml:name>B1</gml:name>
<gml:boundedBy>
<gml:Envelope srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
<gml:pos srsDimension="3">5618686.0 2573988.4 158.0</gml:pos>
<gml:pos srsDimension="3">5618705.5 2574049.8 158.2</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<tran:function>B1303</tran:function>
<noise:noiseRoadSegmentProperty>
<noise:NoiseRoadSegment gml:id="CNRS_0815">
<gml:boundedBy>
<gml:Envelope srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
<gml:pos srsDimension="3">5618686.0 2573988.4 158.0</gml:pos>
<gml:pos srsDimension="3">5618705.5 2574049.8 158.2</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<noise:mDay uom="kfzph">2564.123</noise:mDay>
<noise:mEvening uom="kfzph">145.123</noise:mEvening>
<noise:mNight uom="kfzph">1231.123</noise:mNight>
<noise:mDay16 uom="kfzph">2010.123</noise:mDay16>
<noise:pDay uom="percent">25.123</noise:pDay>
<noise:pEvening uom="percent">35.123</noise:pEvening>
<noise:pNight uom="percent">45.123</noise:pNight>
<noise:pDay16 uom="percent">30.123</noise:pDay16>
<noise:dtv uom="kfzp24h">20564.123</noise:dtv>
<noise:speedDayPkw uom="kmph">130.123</noise:speedDayPkw>
<noise:speedEveningPkw uom="kmph">100.123</noise:speedEveningPkw>
<noise:speedNightPkw uom="kmph">50.123</noise:speedNightPkw>
<noise:speedDayLkw uom="kmph">80.123</noise:speedDayLkw>
<noise:speedEveningLkw uom="kmph">80.123</noise:speedEveningLkw>
<noise:speedNightLkw uom="kmph">50.123</noise:speedNightLkw>
<noise:roadSurfaceMaterial>Pflaster mit ebener
Oberfläche</noise:roadSurfaceMaterial>
<noise:roadSurfaceCorrection uom="dB">2.123</noise:roadSurfaceCorrection>
<noise:distanceCarriageway uom="m">15.123</noise:distanceCarriageway>
<noise:distanceD uom="m">10.123</noise:distanceD>
<noise:bridge>true</noise:bridge>
<noise:tunnel>false</noise:tunnel>
<noise:roadGradientPercent uom="percent">5.245</noise:roadGradientPercent>
```

```
<noise:lod0BaseLine>
<gml:LineString srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783" srsDimension="3">
<gml:coordinates decimal="." cs="," ts=" ">5618686.0, 2573988.4,158.200000
5618692.5,2574008.8,158.000000
5618705.5,2574049.8,158.100000</gml:coordinates>
</gml:LineString>
</noise:lod0BaseLine>
<noise:lineage>ATKIS-LVermA</noise:lineage>
</noise:NoiseRoadSegment>
</noise:noiseRoadSegmentProperty>
…
</tran:Road>
</cityObjectMember>
<cityObjectMember>
<bldg:Building gml:id="UUID_ef6e19e3-c412-440b-8ba9-24900aa173b5">
<gml:name>small building</gml:name>
<creationDate>2007-01-04</creationDate>
<bldg:function>1060</bldg:function>
<bldg:measuredHeight uom="m">2.38</bldg:measuredHeight>
<bldg:lod1Solid>
<gml:Solid>
<gml:exterior>
<gml:CompositeSurface>
<gml:surfaceMember>
<gml:Polygon srsName=
"urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
<gml:outerBoundaryIs>
<gml:LinearRing>
<gml:coordinates cs="," decimal="." ts=" "
>5662497.03,2559357.47,38.2357750703488
5662489.23,2559355.51,38.2357750703488
5662488.178,2559355.247,38.2357750703488
5662489.022,2559351.872,38.2357750703488
5662497.877,2559354.097,38.2357750703488
5662501.43,2559354.99,38.2357750703488
5662500.584,2559358.357,38.2357750703488
5662497.03,2559357.47,38.2357750703488</gml:coordinates>
</gml:LinearRing>
</gml:outerBoundaryIs>
</gml:Polygon>
</gml:surfaceMember>
…
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</bldg:lod1Solid>
<bldg:address>
<Address>
```

```
<xalAddress>
<xAL:AddressDetails>
<xAL:Country>
<xAL:CountryName>Germany</xAL:CountryName>
<xAL:Locality Type="Town">
<xAL:LocalityName>Musterstadt</xAL:LocalityName>
<xAL:Thoroughfare Type="Street">
<xAL:ThoroughfareNumber>1</xAL:ThoroughfareNumber>
<xAL:ThoroughfareName>Musterstrasse</xAL:ThoroughfareName>
</xAL:Thoroughfare>
<xAL:PostalCode>
<xAL:PostalCodeNumber>10000</xAL:PostalCodeNumber>
</xAL:PostalCode>
</xAL:Locality>
</xAL:Country>
</xAL:AddressDetails>
</xalAddress>
</Address>
</bldg:address>
<noise:buildingReflection>Fassade</noise:buildingReflection>
<noise:buildingReflectionCorrection uom="dB">
3.23</noise:buildingReflectionCorrection>
<noise:buildingLDenMax uom="dB">10</noise:buildingLDenMax>
<noise:buildingLDenMin uom="dB">30</noise:buildingLDenMin>
<noise:buildingLDenEq uom="dB">20</noise:buildingLDenEq>
<noise:buildingLNightMax uom="dB">40</noise:buildingLNightMax>
<noise:buildingLNightMin uom="dB">60</noise:buildingLNightMin>
<noise:buildingLNightEq uom="dB">50</noise:buildingLNightEq>
<noise:buildingHabitants>32</noise:buildingHabitants>
<noise:buildingAppartments>8</noise:buildingAppartments>
<noise:buildingImmissionPoints>45 1 1 1 50 2 2 2
</noise:buildingImmissionPoints>
</bldg:Building>
</cityObjectMember>
</CityModel>
```

Annex I: （资料性附录）ADE在泛在网络机器人服务中的应用示例

本附件说明了CityGML在机器人服务中的使用。以相应的应用领域扩展（ADE）的定义为例。

I.1. 泛在网络机器人概述

在机器人领域，有一个起源于日本的概念叫网络机器人，其目的是通过连接多个机器人来实现先进机器人服务。目标是超越单个机器人的能力极限。嵌入环境中的传感器和互联网上的代理人，也被广泛视为机器人，推动了通过网络连接机器人的网络机器人技术研发。与单个机器人相比，连接多个机器人提供了更好的认知能力和交互能力，人们认为先进的机器人服务是可以实现的。

在日本，网络机器人论坛成立于2003年，其目标是顺利有效地推进网络机器人的研发及其标准化。有大约150个成员，包括机械制造商、电子制造商、电信运营商、研究机构、学术专家、经济组织和地方政府。他们从事网络机器人的研究、推广、开发和标准化等活动。

在国家层面，日本内务省自2009年起开始“面向老年人和残疾人的泛在网络机器人技术（UNR）研发”，日本的泛在网络机器人研发工作正在全面展开。

在泛在网络机器人的研究与开发项目中，网络机器人服务从单点扩展到多点是一个值得考虑的问题。为了扩展网络机器人的活动范围，提供各种机器人服务，需要有一个框架来控制周围环境信息作为空间属性，并以机器人能够理解的方式提供服务。 在这些研究和开发项目中，控制网络机器人空间属性的数据库称为空间主数据库。
基于机器人在建筑物内外移动的假设，在空间主数据库中使用了CityGML模型，该模型能够提供室内外特征的表示。然而，移动机器人还是缺乏必要的信息，例如楼层结构、地板材料和门属性。因此，使用ADE机制对CityGML数据模型进行了扩展。图102示出了空间主数据库3D可视化的示例，该空间主数据库是使用从CityGML生成的Robotics ADE，并且在这些研究项目中使用。

用于移动机器人室内定位的坐标系不是室外使用的全局坐标系，而是为每个建筑物配置的局部坐标系。如果建筑有多个楼层，则可能会为每个楼层应用不同的坐标系。此外，一个楼层可能应用多个坐标系。为了与人类进行有效交互，对每个建筑设置的多个坐标系进行控制，以及使用空间主数据库，对单个对象的语义知识进行管理，变得特别关键。

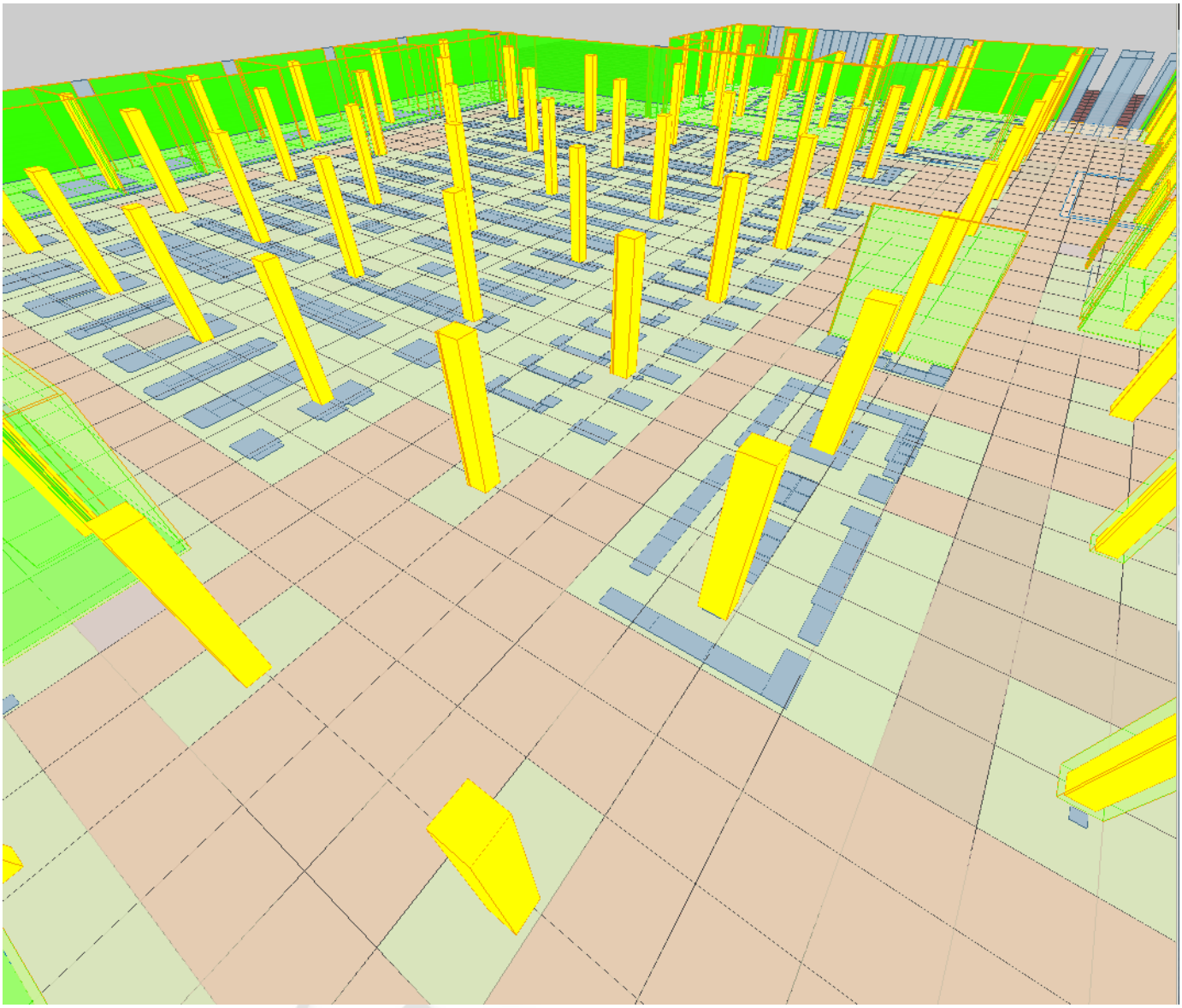

图 102. 购物中心的3D空间主数据库。 这是从CityGML和Robotics ADE生成的，用于日本无处不在的网络机器人服务研究项目。 每个楼层的高度是在空间主数据库中将原始2D CAD数据转换为3D数据时设置的。 因此，某些设备（例如家具）的高度未设置并以2D方式显示。 此外，由空间主数据库的网格地图功能生成的网格地图（带有障碍物信息）附着在地板表面的地面上。 网格地图由给定大小的网格单元组成，并根据障碍物是否存在进行颜色编码。机器人使用这些网格图来识别它们是否可以通过每个网格（绿色=墙壁，黄色=柱子，蓝色=设备，绿色地板=有障碍物（例如柱子和家具）的网格单元，红色地板=没有障碍物的网格单元）。

在研发项目中，2010年在一家商业设施进行了演示实验。图103显示了在一家购物中心进行的实验。机器人提供的服务包括与顾客交谈、带顾客参观商店、协助购物等。这个购物中心设施由多个建筑物组成，测试了机器人在建筑物内外移动并带领顾客参观商店的服务。

空间主数据库信息被转换成栅格地图（光栅地图）格式，并由服务器提供给机器人。栅格地图是PNG图像格式，支持四种栅格地图类型：障碍物、地板材质、地板坡度和地板凹凸纹理。网格图是将地板表面划分为给定大小的网格单元，并对每个网格单元的信息进行编码。每个网格单元的信息以网格地图图像格式，并基于不同颜色像素进行显示。该网格单元需要有关地板材料、地板坡度和地板凹凸的信息。取决于机器人移动机构的功能和限制，上述信息可能成为机器人无法通过的障碍。 在用于演示实验的商场中，建筑物之间的走廊存在高角度的斜坡。实验中使用了两种不同类型的机器人，一种能爬坡，另一种不能。后一个机器人通过网络呼叫前一个机器人，当它到达网格地图上所示的不可通行区域时，它就将客户指南交给了前一个机器人。

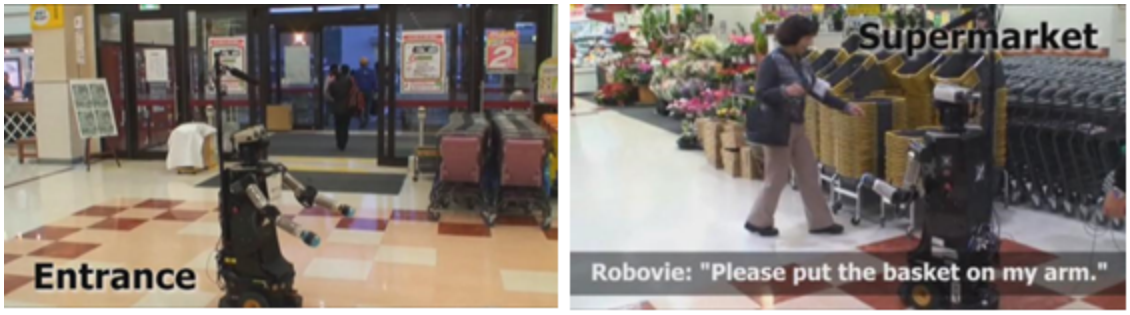

图 103. 使用机器人（日本ATR的Robovie-II）在购物中心进行的演示实验中的场景（来源：“Robovie II 在超市帮助老年顾客”，©ATR（国际高级电信研究所）。

I.2. 空间主数据库概述

本节对用于网络机器人的空间主数据库进行概述。

在建立空间主数据库的数据模型时，考虑了与CityGML LOD4的兼容性。此外，还添加了移动机器人所必需的地板材料（*materialType*）、*Storey*和门属性（*doorOperationType*）的数据。第I.3节提供了更多细节。

图104示出了在第I.1节中描述的演示实验中，使用的空间主数据库管理系统的配置。该系统将建筑物的信息，如CAD数据或CityGML数据导入到空间主数据库中。当将其转换为栅格地图时，该系统搜索目标三维对象并将其投影到二维地板表面。下一步是将其划分为网格，指定楼层编号、楼层面积（边界框）和网格大小（宽度和长度）来划分楼层表面。最后，生成与每个网格信息相关的PNG图像（信息以像素值形式表示），例如地板材料或是否存在障碍物。将三维模型转换为网格地图后，系统将其提供给区域管理网关服务器。

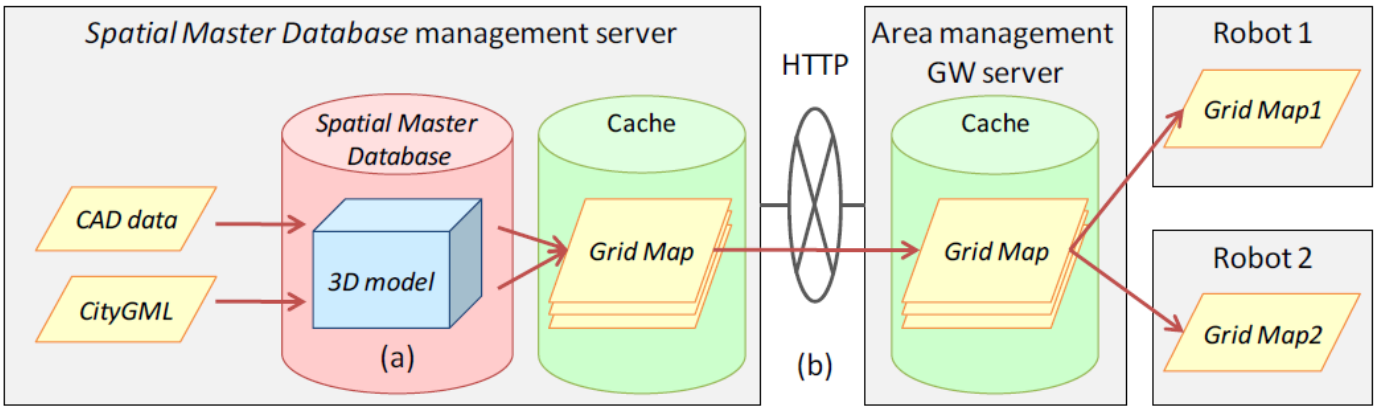

图 104. 空间主数据库管理系统的配置。

进行演示实验的购物中心由两座建筑物（室内）和一个连接建筑物的走廊（室外）组成（如图105所示）。空间主数据库主要存储室内空间数据、走廊室外空间数据。图106示出了基于地板材料，生成的栅格图示例。基于WMS（仓库管理系统）接口提供网格地图服务。其中，WMS接口基于“室外和室内3D路由服务工程报告”进行开发（cf. Sato 2009, OGC Doc. No. 09-067r2）。

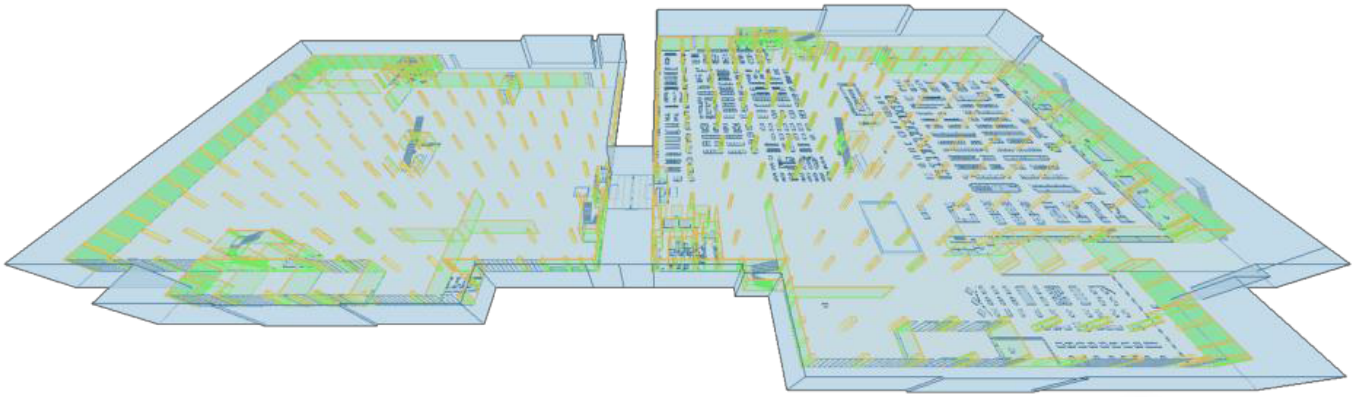

图 105. 购物中心的 3D 模型，由两座建筑物（室内）和连接建筑物的走廊（室外）组成。

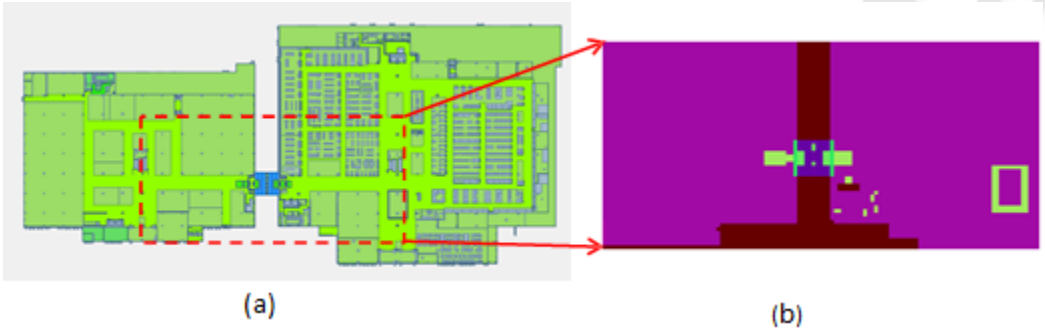

图 106. 当网格图是地板材料时，从空间数据 (a) 生成网格图 (b) 的示例。

I.3. CityGML ADE概述

本节介绍如何使用CityGML实现第I.2节中介绍的空间主数据库。具体来说，本节展示了作为机器人应用程序架构，其数据模型的UML图和XML模式。该机器人应用程序架构是基于两组ADE实现的。之所以要准备两个ADE，是因为泛在网络机器人服务需要两种类型的信息。一个ADE是提供一般信息，它不仅可用于机器人服务，也可用于室内服务。另一个是关于机器人服务的信息。第一个ADE是*CityGML Standard Opening ADE*，基于CityGML扩展的*_Opening*、*Door*和*Window*类的架构。第二个ADE是*UNR（Ubiquitous Network Robots）ADE*，它是一个添加类（如*Storey*）和属性（如门类型和地板材质）的架构，上述类泛在网络机器人是必需的。

本节的目的是提供一个示例，说明如何使用多个AED扩展CityGML。由于特定属性和对象类型（作为代码表实现）的语义来自日本国家泛在网络机器人服务研究项目，因此这里不详细解释它们。本节介绍了标准开口ADE和UNR ADE。ADE的详细信息在CityGML Wiki中进行了解释（参见http://www.citygmlwiki.org/index.php/CityGML-ADEs). 两个ADE的XML模式定义以及示例数据集和代码列表可以从下方链接中进行查看 http://schemas.opengis.net/citygml/examples/2.0/ade/robotics-ade/ 。

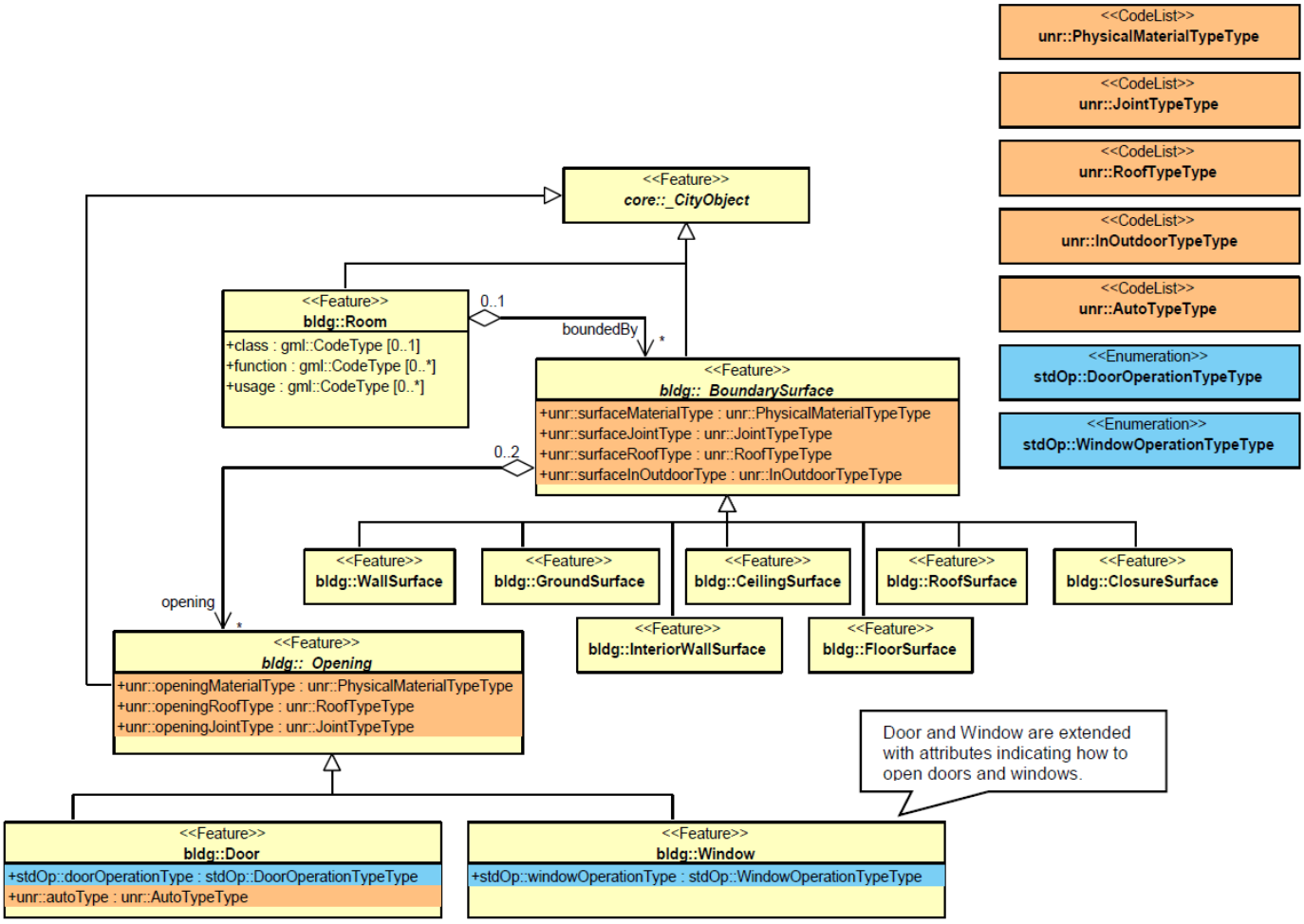

图 107. CityGML Robotics 应用模式 - 标准开口和 UNR 模型，包括门窗（浅黄色 = CityGML 模块，浅蓝色 = CityGML 标准开口 ADE，浅橙色 = UNR ADE）。前缀用于指示与模型元素关联的 XML 命名空间。前缀stdOp与CityGML 标准开放 ADE（来源：柏林技术大学大地测量和地理信息科学研究所）相关联，前缀unr与 UNR ADE 相关联（来源：中央研究实验室，日立有限公司）。CityGML 模型元素与推荐的 CityGML 前缀相关联。

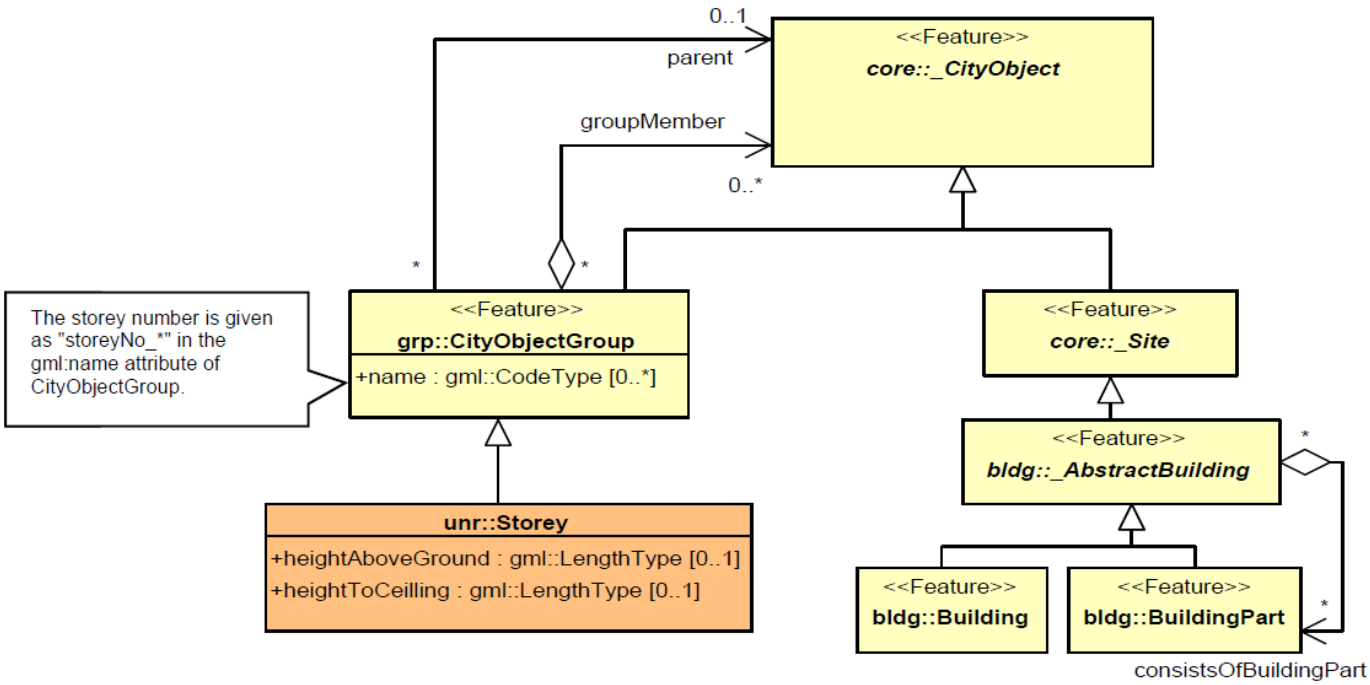

图 108. CityGML Robotics 应用模式——UNR Storey 模型（浅黄色=CityGML 模块，浅橙色=UNR ADE）。 前缀用于指示与模型元素关联的 XML 命名空间。 前缀unr与 UNR ADE 相关联（来源：Hitachi, Ltd. 中央研究实验室）。 CityGML 模型元素与推荐的 CityGML 前缀相关联。

标准架构定义的文件头

```
<xsd:schema xmlns="http://unr.crl.hitachi.co.jp/ade/standard_opening"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:gml=
"http://www.opengis.net/gml" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0"
targetNamespace="http://unr.crl.hitachi.co.jp/ade/standard_opening"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xsd:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xsd:import namespace="http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://schemas.opengis.net/citygml/building/2.0/building.x
sd"/>
…
</xsd:schema>
```

Door的应用程序特定属性

```
<xsd:element name="doorOperationType" type="DoorOperationTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfDoor"/>
<xsd:simpleType name="DoorOperationTypeType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="swinging"/>
<xsd:enumeration value="double_acting"/>
<xsd:enumeration value="sliding"/>
<xsd:enumeration value="folding"/>
<xsd:enumeration value="revolving"/>
<xsd:enumeration value="rollingup"/>
<xsd:enumeration value="userdefined"/>
<xsd:enumeration value="notdefined"/>
</xsd:restriction>
</xsd:simpleType>
```

Window的应用程序特定属性

```
<xsd:element name="windowOperationType" type="WindowOperationTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfDoor"/>
<xsd:simpleType name="WindowOperationTypeType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="sidehungrighthand"/>
<xsd:enumeration value="sidehunglefthand"/>
<xsd:enumeration value="tiltandturnrighthand"/>
<xsd:enumeration value="tiltandturnlefthand"/>
<xsd:enumeration value="tophung"/>
<xsd:enumeration value="bottomhung"/>
<xsd:enumeration value="pivothorizontal"/>
<xsd:enumeration value="pivotvertical"/>
<xsd:enumeration value="slidinghorizontal"/>
<xsd:enumeration value="slidingvertical"/>
<xsd:enumeration value="removablecasement"/>
<xsd:enumeration value="fixedcasement"/>
<xsd:enumeration value="otheroperation"/>
<xsd:enumeration value="notdefined"/>
</xsd:restriction>
</xsd:simpleType>
```

UNR ADE架构定义的文件头

```
<xsd:schema xmlns="http://unr.crl.hitachi.co.jp/ade/unr" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:core="http://www.opengis.net/citygml/2.0" xmlns:bldg=
"http://www.opengis.net/citygml/building/2.0" xmlns:grp=
"http://www.opengis.net/citygml/cityobjectgroup/2.0"
targetNamespace="http://unr.crl.hitachi.co.jp/ade/unr" elementFormDefault
="qualified" attributeFormDefault="unqualified">
<xsd:import namespace="http://www.opengis.net/gml" schemaLocation=
"http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<xsd:import namespace="http://www.opengis.net/citygml/building/2.0"
schemaLocation="http://schemas.opengis.net/citygml/building/2.0/building.x
sd"/>
<xsd:import namespace="http://www.opengis.net/citygml/cityobjectgroup/2.0"
schemaLocation="http://schemas.citygml.org/citygml/cityobjectgroup/2.0/cit
yObjectGroup.xsd"/>
…
</xsd:schema>
```

UNR StoreyType, Storey

```
<xsd:complexType name="StoreyPropertyType">
<xsd:sequence minOccurs="0">
<xsd:element ref="Storey"/>
</xsd:sequence>
<xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xsd:complexType>
<xsd:element name="storeyProperty" type="StoreyPropertyType"
substitutionGroup="grp:_GenericApplicationPropertyOfCityObjectGroup"/>

<xsd:complexType name="StoreyType">
<xsd:complexContent>
<xsd:extension base="grp:CityObjectGroupType">
<xsd:sequence>
<xsd:element name="heightAboveGround" type="gml:LengthType" minOccurs="0"
maxOccurs="1"/>
<xsd:element name="heightToCeiling" type="gml:LengthType" minOccurs="0"
maxOccurs="1"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="Storey" type="StoreyType" substitutionGroup=
"grp:CityObjectGroup"/>
```

_Opening的应用程序特定属性

```
<xsd:element name="openingMaterialType" type="PhysicalMaterialTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfOpening"/>
<xsd:element name="openingRoofType" type="RoofTypeType" substitutionGroup
="bldg:_GenericApplicationPropertyOfOpening"/>
<xsd:element name="openingJointType" type="JointTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfOpening"/>
```

Door的应用程序特定属性

```
<xsd:element name="autoType" type="AutoTypeType" substitutionGroup=
"bldg:_GenericApplicationPropertyOfDoor"/>
```

_BoundarySurface的应用程序特定属性

```
<xsd:element name="surfaceMaterialType" type="PhysicalMaterialTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfBoundarySurface"/>
<xsd:element name="surfaceRoofType" type="RoofTypeType" substitutionGroup
="bldg:_GenericApplicationPropertyOfBoundarySurface"/>
<xsd:element name="surfaceJointType" type="JointTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfBoundarySurface"/>
<xsd:element name="surfaceInOutdoorType" type="InOutdoorTypeType"
substitutionGroup="bldg:_GenericApplicationPropertyOfBoundarySurface"/>
```

I.4. 示例数据集

以下数据集说明了使用Robotics应用模式的CityGML实例文档。它包含三个_CityObject要素：具有*Storey*对象属性的*CityObjectGroup*对象、具有材料类型属性的*FloorSurface*对象和具有*doorOperationType*属性的*Door*对象。该数据集引用了Robotics ADE的XML模式定义文件。该文件特地导入了由CityGML标准Opening ADE（扩展的模块为*CityGML Core*和*Building*模块）和UNR ADE（扩展的模块为*CityGML Core*,*Building*和*CityObjectGroup*模块）共同扩展的CityGML模块的XML模式定义。因此，被采用的CityGML模块所定义的所有类都可以在实例文档中使用。此外，还可以使用特定应用程序的添加项，例如新的对象类型（例如*Storey*），和附加主题属性（例如为*Door*定义的属性）。这些附加元素与标准CityGML元素的不同之处在于，前者的命名空间的前缀*stdOp*和*unr*指代的是Robotics模式定义。

为了使用多个ADE文件，需要遵循XML模式规则，并在*xsi:schemaLocation*标签中明确指定要读取多个（XML模式的）ADE文件。在以下数据集中，两对模式位置，标准开放ADE和UNR ADE，在*xsi:schemaLocation*中被编码。当使用类似于CityGML解析器等软件的时候，必须读取多个XML模式文件。此外，当使用多个ADE文件时，可能会出现循环引用等问题，应尽量避免。

列表15：摘自使用了被图示说明过的CityGML Robotics应用程序模式的CityGML数据集。请参阅模式中包含的代码列表，以获取有关属性值（例如unr:openingMaterialType对应2491）的信息。

```
<?xml version="1.0" encoding="UTF-8"?>
<CityModel xmlns="http://www.opengis.net/citygml/2.0" xmlns:xlink=
```

```
"http://www.w3.org/1999/xlink" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:gml=
"http://www.opengis.net/gml"
xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:grp=
"http://www.opengis.net/citygml/cityobjectgroup/2.0" xmlns:stdOp=
"http://unr.crl.hitachi.co.jp/ade/standard_opening" xmlns:unr=
"http://unr.crl.hitachi.co.jp/ade/unr"
xsi:schemaLocation="http://unr.crl.hitachi.co.jp/ade/standard_opening
http://unr.crl.hitachi.co.jp/ade/standard_opening/stdOp.xsd
http://unr.crl.hitachi.co.jp/ade/unr
http://unr.crl.hitachi.co.jp/ade/unr/unr.xsd">
<cityObjectMember>
<grp:CityObjectGroup gml:id="Storey_0">
<gml:name>Sample Storey 0</gml:name>
<gml:name>storeyNo_0</gml:name>
<grp:class>building separation</grp:class>
<grp:function>lod4Storey</grp:function>
<grp:groupMember xlink:href="#FloorSurface_1"/>
<grp:groupMember xlink:href="#Door_1"/>
<unr:storeyProperty>
<unr:Storey>
<unr:heightAboveGround uom="#m">0.0</unr:heightAboveGround>
<unr:heightToCeiling uom="#m">5.0</unr:heightToCeiling>
</unr:Storey>
</unr:storeyProperty>
</grp:CityObjectGroup>
</cityObjectMember>

<cityObjectMember>
<bldg:Building>
…
<bldg:boundedBy>
<bldg:FloorSurface gml:id="FloorSurface_1">
<gml:name>Sample FloorSurface 1</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3"> 0.0 0.0 0.0 50.0 0.0 0.0 50.0 50.0 0.0 0.0
50.0 0.0 0.0 0.0 0.0
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
```

```
<unr:surfaceMaterialType>2491</unr:surfaceMaterialType>
<unr:surfaceRoofType>2</unr:surfaceRoofType>
<unr:surfaceInOutdoorType>2</unr:surfaceInOutdoorType>
<unr:surfaceJointType>4</unr:surfaceJointType>
</bldg:FloorSurface>
</bldg:boundedBy>
…
<bldg:boundedBy>
<bldg:WallSurface>
…
<bldg:opening>
<bldg:Door gml:id="Door_1">
<gml:name>Sample Door 1</gml:name>
<bldg:lod4MultiSurface>
<gml:MultiSurface>
<gml:surfaceMember>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="3"> 0.0 0.0 0.0 0.0 0.0 20.0 0.0 10.0 20.0 0.0
10.0 0.0 0.0 0.0 0.0 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:surfaceMember>
</gml:MultiSurface>
</bldg:lod4MultiSurface>
<unr:openingMaterialType>2491</unr:openingMaterialType>
<unr:openingRoofType>2</unr:openingRoofType>
<unr:openingJointType>1</unr:openingJointType>
<stdOp:doorOperationType>swinging</stdOp:doorOperationType>
</bldg:Door>
</bldg:opening>
</bldg:WallSurface>
</bldg:boundedBy>
…
</bldg:Building>
</cityObjectMember>
</CityModel>
```

Annex J: （资料性附录）修订历史

日期	发布版本	编辑	描述
2006-02-01	0.1.0	Czerwinski, Kolbe, Gröger	启动CityGML标准项目
2006-02-22	0.1.0	Gröger, Gruber	补充UML图
2006-04-26	0.2.0	Kolbe, Gröger, Czerwinski	向欧洲SDR发布CityGML标准草案
2006-03-06	0.3.0	Kolbe, Gröger, Czerwinski	更改属性名称，添加部分属性，向OGC发布CityGML标准文件
2007-05-30	0.4.0	Kolbe, Gröger, Nagel, Lorenz, Benner,Czerwinski,Gruber, Schlüter, Bildstein, Drees, Löwner	引入全新的外观模型、应用领域扩展（ADE）简介、修改小部分建筑模型、修改小部分城市对象组、在城市家具中增加了地形相交曲线的概念、代码列表应用，向OGC发布CityGML标准文件
2008-02-04	1.0.0	Kolbe, Gröger, Nagel, Stadler, Lorenz	介绍CityGML数据模型模块化、修改部分外观模型、修改部分城市对象组和交通对象、外部代码列表的编码已更改为GML 3.1.1简单字典专用文件、修订UML图，向CityGML 1.0 SWG发布CityGML标准文件草案
2008-05-19	1.0.0	Kolbe, Gröger, Nagel, Stadler, Lorenz	根据OGC RFC公示阶段的意见和建议，以及CityGML 1.0 SWG的讨论，对部分内容进行变更和修改，CityGML标准文件草案最终发布至CityGML 1.0 SWG
2008-08-18	1.0.0	Carl Reed	对OGC™ 标准的发布进行准备，并进行最终编辑

日期	发布版本	编辑	描述
2011-07-19	1.1.0	Nagel, Häfele, Benner, Gröger, Gruber, Schlüter	向CityGML SWG首次发布CityGML标准文件草案
2011-11-21	1.1.0	Nagel, Häfele, Gröger, Kolbe	根据OGC RFC公示阶段的意见和建议，以及CityGML SWG的讨论，对部分内容进行变更和修改，CityGML标准文件草案最终发布至CityGML SWG
2012-02-27	2.0.0	Nagel, Häfele, Gröger, Kolbe	版本从1.1.0更改为2.0.0，以符合135r11中定义的OGC版本控制策略。为反映主要版本号的更改，将本文件的参考号更改为12-019。向CityGML SWG发布CityGML标准文件草案。
2012-03-14	2.0.0	Carl Reed	对OGC标准进行最终编辑

Annex K: 参考文献

Albert, J., Bachmann, M., Hellmeier, A (2003): Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodelle. Erhebungen im Rahmen der SIG 3D der GDI NRW (in German only). Available at http://www.ikg.uni-bonn.de/fileadmin/sig3d/pdf/Tabelle_Anwendungen_Zielgruppen.pdf.

Becker, T., Nagel, C., Kolbe, T.H. (2011): Integrated 3D modeling of multi-utility networks and their interdependencies for critical infrastructure analysis. In: Kolbe, T.H., König, G., Nagel, C. (Eds.), Advances in 3D Geo-Information Sciences. Springer, Berlin, Heidelberg, pp. 1–20.

Benner, J., Geiger, A., Leinemann, K. (2005): Flexible generation of semantic 3D building models. In: Gröger, G., Kolbe, T. H. (eds.): First International ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models. Bonn, Germany, EuroSDR Publication No. 49. pp.17-22.

Berlo, L. van, Laat, R. de, (2011): Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In: Kolbe, T.H., König, G., Nagel, C. (Eds.), Advances in 3D Geo-Information Sciences. Spring-er, Berlin, Heidelberg, pp. 211–225.

BImSchV (2006): Federal Government of Germany (eds.): 34. Verordnung zur Durchführung des Bundesimmis-sionsschutzgesetzes - Verordnung über die Lärmkartierung [34. BImSchV], in der Fassung vom 6.3.06, in Kraft seit 16.3.06, In: BGBl. I 06, Nr. 12, S. 516 (in German only).

Bleifuß, R., Donaubauer, A., Liebscher, J., Seitle, M., (2009): Entwicklung einer CityGML-Erweiterung für das Facility Management am Beispiel Landeshauptstadt München. In: Symposium angewandte Geoinformatik (AGIT), Salzburg. Booch, G., Rumbaugh, J., Jacobson, I. (1997): Unified Modeling Language User Guide. Addison-Wesley.

CityGML 2007. Official homepage of CityGML. http://www.citygml.org.

Cox, S., Daisy, P., Lake, R., Portele, C., Whiteside, A. (2004): OpenGIS Geography Markup Language (GML 3.1), Implementation Specification Version 3.1.0, Recommendation Paper, OGC Doc. No. 03-105r1.

Czerwinski, A., Kolbe, T. H., Plümer, L., Stöcker-Meier, E. (2006): Interoperability and accuracy requirements for EU environmental noise mapping, In: Kremers, Horst (ed.): Proceedings, InterCarto – InterGIS 12. Ber-lin 2006.

Czerwinski, A., Kolbe, T. H., Plümer, L., Stöcker-Meier, E. (2006b): Spatial data infrastructure techniques for flexible noise mapping strategies, In: Tochtermann, K. / Scharl, A. (eds.): Proceedings of the 20th International Conference on Environmental Informatics - Managing Environmental Knowledge. Graz 2006.

Czerwinski, A., Sandmann, S., Stöcker-Meier, E., Plümer, L. (2007): Sustainable SDI for EU noise mapping in NRW – best practice for INSPIRE, In: International Journal for Spatial Data Infrastructure Research (IJSDIR, http://ijsdir.jrc.it), 2007, vol. 1.

Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., Teichmann, K. (2006): The Virtual 3D City Model of Berlin - Managing, Integrating and Communicating Complex Urban Information. In: Proc. of the 25th Intern. Symposium on Urban Data Management UDMS 2006 in Aal-borg, Denmark, 15-17 May 2006.

European Union (eds.): Directive 2002/49/EG of the European Parliament and of the Council of 25 June

2002 relating to the assessment and management of environmental noise, in: Official Journal of the European Communities from 18.7.02.

Federal Government of Germany (eds.): Gesetz zur Umsetzung der EG-Richtlinie über die Bewertung und Bekämpfung von Umgebungslärm vom 24. Juni 2005, in: Bundesgesetzblatt Jg. 2005, Teil 1, Nr. 38, Bonn 29.6.05 (in German only). Available at http://217.160.60.235/BGBL/bgbl1f/bgbl105s1794.pdf.

Foley, J., van Dam, A,. Feiner, S., Hughes, J. (1995): Computer Graphics: Principles and Practice. Addison Wesley, 2nd Ed.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.

Gröger, G., Plümer, L. (2005): How to get 3-D for the Price of 2-D – Topology and Consistency of 3-D Urban GIS. Geoinformatica, 9(2).

Gröger, G., Benner, J., Dörschlag, D., Drees, R., Gruber U., Leinemann K., Löwner, M.-O.(2005): Das interope-rable 3D-Stadtmodell der SIG 3D, in: Zeitschrift für Vermessungswesen (in German only).

Herring, J. (2001): The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Sche-ma), Version 5. OGC Document Number 01-101.

IAI (2006): International Alliance for Interoperability: IFC 2x3 – Industry Foundation Classes. http://www.iai-international.org/Model/IFC(ifcXML)Specs.html, 2006.

Khronos Group Inc., Sony Computer Inc. (June 2006): COLLADA – Digital Asset Schema Release 1.4.1 – Specification. http://www.khronos.org/files/collada_spec_1_4.pdf.

Kohlhaas, A., Mitchell, J. (2007): Modelling cities. In: Constructing the future: nD modelling, p. 372-392, Taylor & Francis

Kolbe, T. H., Gröger, G. (2003): Towards unified 3D city models. In: Schiewe, J., Hahn, M., Madden, M., Sester, M. (eds): Challenges in Geospatial Analysis, Integration and Visualization II. Proc. of Joint ISPRS Workshop, Stuttgart.

Kolbe, T. H., Gröger, G., Plümer, L. (2005): CityGML – Interoperable Access to 3D City Models, In: van Oosterom, Peter, Zlatanova, Sisi, Fendel, E.M. (eds.): Geo-information for Disaster Management. Proc. of the 1st International Symposium on Geo-information for Disaster Management, Delft, The Netherlands, March 21-23. Delft.

Kolbe, T. H., Gröger, G., Plümer, L. (2008): CityGML - 3D City Models for Emergency Response, In: Zlatano-va, Li (eds.): Geospatial Information Technology for Emergency Response, ISPRS book series, Taylor & Francis.

Kolbe, T. H., Nagel, C., Lorenz, A. (2009): CityGML – OGC Standard for Photogrammetry?, In: Fritsch, D. (ed.): Photogrammetric Week, Wichmann, Heidelberg, pp. 265-277.

OASIS 2003: xNAL Name and Address Standard. Organization for the Advancement of Structured Information Standards. http://xml.coverpages.org/xnal.html

Okabe, A., Boots, B., Sugihara, K. (1992): Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. John Wiley & Sons.

Sato, A. (ed.) (2009): OWS-6 Outdoor and Indoor 3D Routing Services Engineering Report, Public Engineering Report, OGC Doc. No. 09-067r2.

Schulte, C., Coors, V., (2008): Development of a CityGML ADE for dynamic 3D flood information,. In: Joint ISCRAM-CHINA and GI4DM Conference on Information Systems on Information Systems for Crisis Management, Harbin, China, 2008.

Stadler, A., Kolbe, T. H. (2007): Spatio-Semantic Coherence in the Integration of 3D City Models. In: Proceed-ings of 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede, The Nether-lands, 13-15 June 2007

VBUS (2006): Federal Government of Germany (eds.): Vorläufige Berechnungsmethode für den Umgebungs-lärm an Straßen vom 15.5.2006 (in German only).

VBUSch (2006): Federal Government of Germany (eds.): Vorläufige Berechnungsmethode für den Umgebungs-lärm an Schienenwegen vom 10.5.2006 (in German only).

VRML97, Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification and UTF-8 encoding. Part 1 of ISO/IEC Standard 14772-1:1997.

Web 3D, Information technology - Computer graphics and image processing - Extensible 3D (X3D). ISO/IEC 19775:2004. http://www.web3d.org/x3d/specifications/, 2004.

Whiteside, A. (ed.) (2009): Definition identifier URNs in OGC namespace, Version: 1.3, OGC® Best Practices, OGC Doc. No. 07-092r3.

Whiteside, A. (ed.) (2005): GML 3.1.1 Simple Dictionary Profile, Version: 1.0.0, OpenGIS® Implementation Specification, OGC Doc. No. 05-099r2.