

Spezifikation von Prüfplänen und Prüfergebnissen zur Validierung von 3D-Stadtmodellen

DETLEV WAGNER¹, THOMAS H. KOLBE² & VOLKER COORS³

Die Datenqualität von 3D-Stadtmodellen gewinnt mit der wachsenden Vielfalt der Anwendungsszenarien zunehmend an Bedeutung. Automatisierte Prüftools sind vorhanden, z.B. CityDoctor oder val3dity, jedoch bestehen noch keine allgemein anerkannten Grundsätze, wie Prüfanforderungen und Validierungsergebnisse dokumentiert werden. Sie sollen sowohl menschenlesbar als auch maschineninterpretierbar sein, damit komplexe Objekte referenziert und in einem späteren Bearbeitungsschritt automatisch korrigiert oder zur Analyse in geeigneter Weise visualisiert werden können.

In diesem Beitrag werden Aufbau und Spezifikation eines Prüfplans beschrieben, der anwendungsspezifische Erfordernisse, wie etwa Geschlossenheit von Volumenkörpern (Wasserdichtigkeit) oder die Verwendung bestimmter Geometriearten berücksichtigt. Ein Modell zur standardisierten Dokumentation und Speicherung von Prüfergebnissen wird vorgestellt. Damit kann die Eignung des Datensatzes für die vorgesehene Anwendung bewertet werden. Schließlich wird ein Verfahren beschrieben, das mittels eines Hashcodes sicherstellt, dass der validierte Datensatz und das Prüfergebnis zusammengehören und konsistent sind. Damit ist eine eindeutige und sichere Zuordnung von geprüftem Datensatz und Prüfergebnis auch zu einem späteren Zeitpunkt gewährleistet.

1 Einleitung

Die Bedeutung der Qualität von 3D-Geodaten steigt mit zunehmender Komplexität der Anwendungsszenarien. Stand bisher meist eine ansprechende Visualisierung im Vordergrund, verschieben sich die Anwendungsfälle mehr und mehr in Richtung Analyse und Simulation, wobei 3D-Stadtmodelle einen wichtigen Teil der Input-Daten darstellen. Beispielhaft seien hier Lärmkataster oder Solarpotentialanalysen angeführt (Czerwinski and Plümer, 2008), (Kaden et al., 2012).

Anwendungsspezifische Anforderungen an die Ausgangsdaten müssen zu deren fehlerfreier Weiterverarbeitung berücksichtigt werden. Dies können z.B. geschlossene Volumenkörper zur Berechnung des Wärmeenergiebedarfs eines Gebäudes sein.

Bisher liegt keine einfache Form der Spezifikation vor, die die Anforderungen an ein Modell in allgemein anerkannter Weise beschreibt. Zudem ist derzeit unklar, wie ein Modell auf die Einhaltung dieser Anforderungen nachvollziehbar überprüft werden kann. Zwar gibt es mit dem CityDoctor (Wewetzer et al., 2013) sowie val3dity (Zhao et al., 2014) erste Prüftools, die bestimmte vordefinierte Modelleigenschaften prüfen können. Deren Konfigurierbarkeit ist

1) Detlev Wagner, Hochschule für Technik Stuttgart, Schellingstraße 24, 70174 Stuttgart;
E-Mail: detlev.wagner@hft-stuttgart.de

2) Thomas H. Kolbe, Technische Universität München, Arcisstraße 21, 80333 München;
E-Mail: thomas.kolbe@tum.de

3) Volker Coors, Hochschule für Technik Stuttgart, Schellingstraße 24, 70174 Stuttgart;
E-Mail: volker.coors@hft-stuttgart.de

jedoch eingeschränkt und insbesondere für den Anwender schwierig zu durchschauen. Das hängt auch mit der uneinheitlichen Verwendung von Begriffen zusammen, die es erschweren, einen allgemein verständlichen Prüfplan zu spezifizieren. Hinzu kommt die Komplexität des Themas Datenqualität in Zusammenhang mit 3D-Daten, deren Feinheiten eine vertiefte Kenntnis des zu Grunde liegenden Datenmodells (meist CityGML) verlangt, die von Anwendern kaum erwartet werden kann.

Im Folgenden wird ein Entwurf vorgestellt, der Anforderungen an einen Prüfplan für 3D-Stadtmodelle spezifiziert und exemplarisch deren Umsetzung beschreibt. Darauf baut die Prüfung von Daten auf, die gegen diesen Prüfplan validiert werden. Die Anforderungen an ein Fehlermodell, das die Validierungsergebnisse umfassend abbildet, werden herausgearbeitet und anhand einer Beispielimplementierung vorgestellt. Zuletzt wird eine Möglichkeit zur sicheren Kopplung von Validierungsergebnis und geprüftem Datensatz beschrieben, die die Einhaltung der Kriterien gemäß Prüfplan sicherstellt. Damit erübrigt sich bei Datenweitergabe die erneute aufwändige Validierung.

2 Problemstellung

Um Diskrepanzen zu vermeiden, müssen die Anforderungen an einen Datenbestand ebenso festgelegt werden wie die Verfahren, mit denen der Datenbestand geprüft wird. Nur so ist ein eindeutiges Verständnis von Datenqualität herzustellen, welches aufgrund der unterschiedlichen Sicht von Datenhersteller und Datennutzer nicht von vorneherein gegeben ist.

2.1 Konventionelle Spezifikation der Anforderungen

Eine exakte Spezifikation der Daten und ihrer Qualität vermindert das Risiko, dass Daten sich als unbrauchbar erweisen. Ein Datenhersteller ist also gehalten, vollständige Metadaten zu liefern, die insbesondere exakt spezifizierte Angaben zur Qualität enthalten. Nur so kann der Nutzer des Produkts entscheiden, ob es seinen Anforderungen entspricht. Umgekehrt sollte der Auftraggeber eine exakte Spezifikation neu zu erstellender Daten für den Datenhersteller bereitstellen. Für beide Seiten stellt die Spezifikation der Anforderungen an das Produkt und deren Überprüfung somit einen zentralen Bestandteil des Informationsaustausches dar.

Tab. 1: Unterschiedliche Modellierung von CityGML-Elementen in LOD2-Modellen

Element	Modell A	Modell B	Modell C
<i>Building</i>	<i>lod2SolidGeometry</i>	<i>lod2MultiSurfaceGeometry</i>	<i>keine eigene Geometrie</i>
<i>BuildingPart</i>	<i>lod2SolidGeometry</i>	<i>lod2MultiSurfaceGeometry</i>	<i>lod2SolidGeometry</i>
<i>BuildingInstallation</i>	<i>lod2MultiSurfaceGeometry</i>	<i>lod2MultiSurfaceGeometry</i>	<i>keine BuildingInstallation</i>
<i>Dachüberstände</i>	<i>lod2MultiSurfaceGeometry</i>	<i>nicht modelliert</i>	<i>lod2MultiSurfaceGeometry</i>
<i>unterirdische Gebäudeteile</i>	<i>nicht berücksichtigt</i>	<i>modelliert, falls Information vorliegt</i>	<i>vollständig modelliert</i>

Bisher ist es geläufig, diese Anforderungen basierend auf bestehenden Standards festzulegen und als Freitext niederzuschreiben. Ein Ergebnis dieser Festlegung könnte etwa so aussehen: Ein Stadtmodell soll in CityGML LoD2 für ein festgelegtes Gebiet neu erstellt werden. Als Ergebnis können sehr unterschiedliche Geometrien entstehen (Tab. 1). Hinzu kommt, dass Geometrie und

Semantik des Stadtmodells kohärent sein sollten. Der geometrischen Struktur des Modells stünde dann eine semantische Entsprechung gegenüber: Nur wenn die Objekte des geometrischen Modells mit Entitäten des semantischen Modells sinnvoll korrespondieren, kann gewährleistet werden, dass geometrische Objekte wissen, "was" sie sind und semantische Entitäten ihre Lage im Raum kennen. Dieser Aspekt ist insbesondere für die Datenintegration wichtig. Unvollständige Entsprechungen bedeuten einen niedrigeren Grad an Kohärenz (Stadler and Kolbe, 2007).

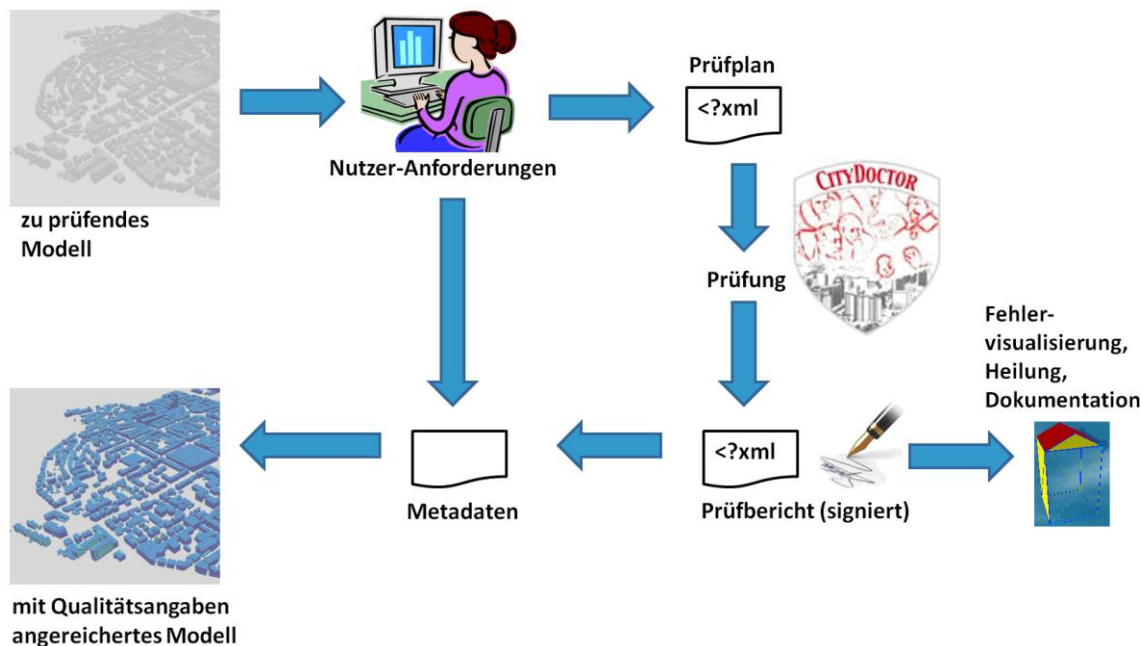


Abb. 1: Bestandteile des Qualitätsprozesses

Je nach Anwendungsbereich können manche Modelle ungeeignet sein, obwohl sie den ursprünglich (zu allgemein) formulierten Anforderungen genügen. Ein Beispiel wäre die Modellierung von Dachgauben, die zwar in der Spezifikation gewünscht wurde, deren Umsetzung jedoch zu unterschiedlichen Modellierungsvarianten führen kann. Wird die Dachgaube etwa als *BuildingInstallation* modelliert anstatt als Teil der Gebäudegeometrie, so führt dies bei der Volumenberechnung zu unterschiedlichen Resultaten. Die Notwendigkeit für eine eindeutige und vollständige Beschreibung der Anforderungen an ein Stadtmodell tritt hier deutlich zu Tage.

2.2 Überprüfung der Anforderungen und Dokumentation des Ergebnisses

Ein Prüfplan beschreibt, wie die Einhaltung der Anforderungen überprüft werden soll. Wegen der hohen Komplexität der Daten und der vielen zu berücksichtigenden Parameter bei der Spezifizierung der einzelnen Anforderungen sollte der Prüfplan in einer standardisierten und automatisch zu verarbeitenden Form erstellt werden.

Das Prüfergebnis (Fehlerprotokoll) beschreibt sodann den Grad der Übereinstimmung der Daten mit den Anforderungen. Das Fehlerprotokoll hängt vom Prüfplan ab und muss eindeutig dem geprüften Datensatz zugeordnet werden können. Ein Vorschlag für Form und Inhalte dieses Prüfplans und des Fehlerprotokolls, sowie einer Signierungsmöglichkeit des Prüfergebnisses,

werden im Folgenden vorgestellt. Eine detaillierte Beschreibung der Geometrieprüfung mit dem CityDoctor findet sich in (Wagner et al., 2013a).

Der Prozess beginnt also mit der Spezifikation der Anforderungen an das Modell. Aus ihnen wird ein formaler Prüfplan erstellt, dessen automatische Verarbeitung einen umfassenden und (sofern gewünscht) signierten Prüfbericht erzeugt. Mit dessen Hilfe lässt sich ein anschließender Heilungsprozess ebenso spezifizieren wie die Anreicherung des geprüften Modells mit Qualitätsinformationen (Abb. 1). Der Prüfbericht kann als Dokument in ein übergeordnetes Qualitätssicherungssystem einfließen.

2.3 Normen und Standards für die Qualität von Geodaten

In der Normenreihe ISO 19100 sind Vorgaben zur Qualität von Geodaten enthalten, die wichtigsten sind ISO 19113 (Geographic Information – Quality principles), ISO 19114 (Geographic Information – Quality evaluation procedures) und ISO/TS 19138 (Geographic Information – Data quality measures).

Die relevanten Elemente zur Beschreibung der Qualität von Geodatensätzen sind gemäß ISO 19113 logische, zeitliche, thematische und Lagegenauigkeit, sowie Vollständigkeit. Die Datenqualitätselemente werden jeweils in mehrere Subelemente differenziert, wovon im Rahmen von 3D-Stadtmodellen topologische und Formatkonsistenz sowie die Korrektheit von Attributwerten bedeutsam sind. Mit Hilfe dieser Systematik lassen sich somit abstrakte Bedingungen für Qualitätsanforderungen definieren.

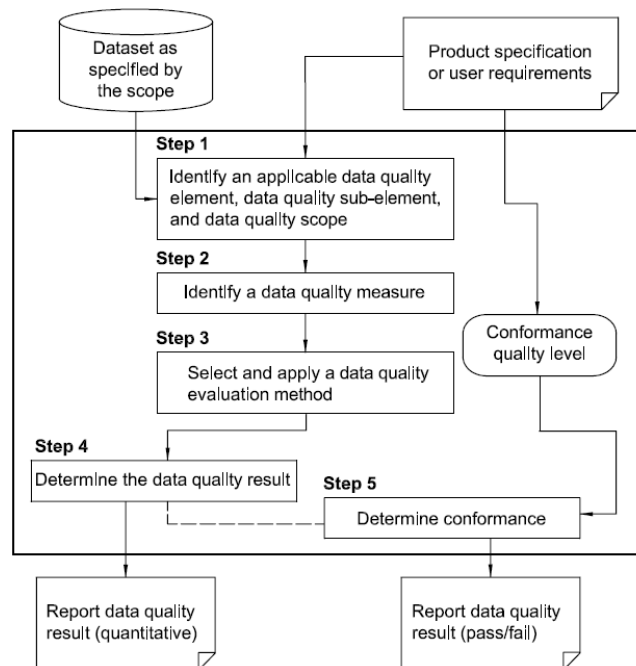


Abb. 2: Ablaufdiagramm des Qualitätsprüfungsprozesses (nach ISO 19114)

ISO 19114 definiert die Elemente des Qualitätsprüfungsprozesses auf abstrakter Ebene (Abb. 2). Der erste Schritt ist dabei die Festlegung der Datenqualitätselemente. Im Unterschied zu klassischen 2D-Geodaten sind die Elemente aus ISO 19113 jedoch nicht hinreichend, um die Eignung eines 3D-Datensatzes für eine bestimmte Anwendung zu gewährleisten. Vielmehr muss

das Qualitätsmodell für diesen Fall erweitert werden, um die Besonderheiten von semantischen 3D-Datenmodellen zu berücksichtigen. Bei der Festlegung der Qualitätselemente sind ein Qualitätsmaß sowie die Methodik zu dessen Prüfung zu definieren. Die Qualitätselemente, die zur Bestimmung der Datenqualität von 3D-Stadtmodellen wichtig sind, werden in Kapitel 3 beschrieben.

3 Prüfplan

Der Prüfplan umfasst die vollständige Spezifikation der Prüfungen, die nötig sind, um einen Datensatz hinsichtlich der einzuhaltenden Anforderungen zu validieren. Dabei ist eine eindeutige und nachvollziehbare Form erforderlich, die von automatisierten Prüfverfahren verarbeitet werden kann.

3.1 Datenqualitätselemente

Die Qualität von 3D-Stadtmodellen wird in großem Maß von Vollständigkeit, korrekter Geometrie und Attributierung, sowie geometrisch-semantischer Konsistenz und Kohärenz bestimmt.

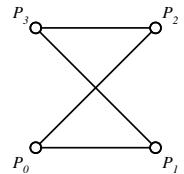
Die Vollständigkeit der Daten kann lediglich gegen externe Daten validiert werden und ist Gegenstand zahlreicher Untersuchungen, beispielsweise (Zielstra and Zipf, 2010), (Kunze et al., 2013). Die weiteren Ausführungen beschränken sich auf eine interne direkte Evaluation der Daten gemäß ISO 19114. Eine Implementierung grundlegender Geometrieprüfungen sowie Plausibilitätschecks von Semantik und Attributen mit Geometriebezug stellt das Validierungstool dar, das im Rahmen des CityDoctor-Projektes in einem gemeinsamen Forschungsvorhaben von HFT Stuttgart und Beuth Hochschule Berlin entstanden ist (Wewetzer et al., 2013).

Die Prüfung der Konsistenz von Geometrie und Semantik erstreckt sich u.a. auf die Art der Modellierung von *BuildingParts* sowie die Plausibilität der *BoundarySurface*-Typen. Attribute mit Geometriebezug können ebenso auf Plausibilität geprüft werden (Wagner et al., 2013b).

Valide Geometrie für 3D-Stadtmodelle besteht im Idealfall aus 2-mannigfaltigen Volumenkörpern, die von planaren Polygonen begrenzt werden (Gröger and Coors, 2010). Wie die dort beschriebenen Regeln exemplarisch in einem Prüfplan umgesetzt werden können, stellen wir im Folgenden am Beispiel der im CityDoctor implementierten Prüfungen dar. Insbesondere können diese in den Kontext der in ISO/TS 19138 gelisteten Standard-Datenqualitätsmaße eingeordnet werden, wobei die meisten dem Bereich der topologischen Konsistenz angehören. Als Beispiel sei hier die Definition der Selbstverschneidungsprüfung von Polygonen in 3D angeführt, die sich anhand der Vorgaben des Maßes darstellen lässt. Zu Grunde liegt diesem Maß die Prüfung linearer Ringe auf Selbstverschneidung, wie sie in Tab. 2 für den CityDoctor definiert ist. Die Zuordnung der Datenqualitätskomponenten gemäß ISO 19138 zu den Bestandteilen der CityDoctor-Definition ist in der rechten Spalte dargestellt.

Der Prüfalgorithmus liefert TRUE, wenn eine Selbstverschneidung gefunden wird und berechnet die notwendigen Rückgabewerte gemäß der Definition, womit ausreichende Informationen zur genauen Inspektion gefundener Fehler sowie für eine automatisierte Heilung vorliegen. Betroffene Geometrielemente werden über eine intern zugewiesene ID referenziert, da diese oft nicht für alle Elemente vorliegt oder erst als Ergebnis der Prüfung erzeugt wird (z.B. für Schnitt

Tab. 2: Definition der Selbstverschneidungsprüfung für Polygone

CityDoctor Check Definition		DQ-Component (ISO 19138)
Name	Keine Selbstschnitte in linearen Ringen	Name
Check-ID	CP_SELFINT	Identifier
CityGML Element	gml:LinearRing	
Beschreibung	Zwei Kanten (P_i, P_{i+1}) und P_k, P_{k+1}) mit $i=0\dots n-1$, $k=0\dots n-1$, $i \neq k$ dürfen sich nur in einem Start-/Endpunkt berühren. Weitere Schnitt- bzw. Berührungspunkte sind nicht zulässig.	Definition/Description
Voraussetzung	CP_NUMPOINTS passed (boolean)	Parameter
Beispiel	 <p>$R = (P_0, P_1, P_3, P_2, P_0)$ Kein valider LinearRing (zwei Kanten schneiden sich)</p>	Example
Prüfergebnis	passed / failed (boolean)	Data quality value type
Rückgabewert	komplex: String: Ring-ID des betroffenen LinearRing List<String>: Edge-IDs der verschneitenen Kanten List <Point>: Point-IDs der Schnittpunkte	Data quality value structure

punkte). Die Summe der gefundenen Fehler ergibt schließlich das Qualitätsmaß *number of invalid self-intersect errors* (D.26) entsprechend den Vorgaben von ISO/TS 19138

Eine vergleichbare Definition nach diesem Muster liegt für alle Prüfungen vor. Zu den reinen Geometrieprüfungen kommen weitere Checks, die sich mit der Konsistenz von Semantik und Geometrie, Teilen der semantischen Modellierung sowie den Attributen beschäftigen (Listing 1). Die Identifikatoren der Prüfungen beginnen mit Präfixen, die auf den Kontext der Prüfung hinweisen. Dabei steht CP_ für Polygonprüfungen, CS_ für Solidprüfungen, SEM_ für Semantik- und C_ für sonstige Prüfungen. Nach einer stark verkürzten Beschreibung folgt der Rückgabewert (in Klammern). Dabei ist zu beachten, dass in jedem Fall die ID des betroffenen Geometrieelements zurückgegeben wird, die in Listing 1 nicht extra angeführt wird. Die Validierung der Geometrie wird als Kombination mehrerer Einzelprüfungen durchgeführt.

Listing 1: In CityDoctor implementierte Prüfungen

- A CP_NUMPOINTS: minimale Punktzahl eines *LinearRing* (LR) (integer)
- CP_CLOSE: LR ist geschlossen
- CP_DUPPOINT: keine Punkt-Dubletten in einem LR (list)
- CP_SELFINT: keine Selbstverschneidungen in einem LR (komplex)
- CP_PLAN: LR muss planar sein (komplex)
- CS_NUMFACES: minimale Seitenanzahl eines *Solids* (integer)
- CS_SELFINT: keine Verschneidung von Polygonen eines *Solids* außerhalb Kanten und Eckpunkten (komplex)
- CS_OUTEREDGE: weniger als zwei inzidente Polygone pro Kante (komplex)
- CS_OVERUSEDGE: mehr als zwei inzidente Polygone pro Kante (komplex)
- CS_FACEORIENT: konsistente Flächenorientierung aller Polygone eines *Solids* (komplex)
- CS_FACEOUT: nach außen weisende Flächennormalen eines *Solids*
- CS_CONCOMP: alle Komponenten eines *Solids* müssen miteinander verbunden sein
- CS_UMBRELLA: alle an einem Punkt inzidenten Polygone können als zusammenhängender Graph dargestellt werden, der sich aus den gemeinsamen Kanten dieser Polygone ergibt (komplex)

- B CS_ISLOD1SOLID: *lod1SolidGeometry* vorhanden
SEM_LOD1_ASSOLID: *lod1SolidGeometry* liegt vor
SEM_LOD1_BUILDINGPARTS: *BuildingParts* sind als *Solid* modelliert
C_MSIFSOLID: *MultisurfaceGeometry* stellt de facto eine Solid-Struktur dar
SEM2_ROOFSURFACE: Flächennormale zeigt nach oben
SEM2_WALLSURFACE: Flächennormale zeigt zur Seite
SEM2_GROUNDSURFACE: Flächennormale zeigt nach unten
C_BS_PLANAR_PATCH: eine Begrenzungsfläche besteht aus nicht-koplanaren Polygonen
C_BS_COPLANAR_SURFACES: eine Begrenzungsfläche besteht aus mehreren koplanaren Polygonen
SEM_LOD1_NUMFLOORS: Das Attribut *storeysAboveGround* ist plausibel bzgl. der Höhe der Geometrie
SEM_GMLID_EXIST: Das Element hat eine GML-ID, die (innerhalb des Datensatzes) eindeutig ist
SEM_MANDATORY_ATTRS: alle Pflichtattribute sind vorhanden
SEM_ATTRIBUTE_DOMAIN: Attribute enthalten gültige Werte aus den entsprechenden Codelisten
SEM_MEASURED_HEIGHT: Das Attribut *measuredHeight* entspricht der Höhe der Gebäudegeometrie
SEM_LOWESTEAVESPOINT: Das Attribut *minimaleTraufhoehe* liegt auf Höhe des niedrigsten Dachpunktes der Geometrie
SEM_SURFACEAREA: Das Attribut *Flaechе* entspricht dem aus der Geometrie berechneten Flächeninhalt

Bei einigen Prüfungen sind Toleranzwerte erforderlich, die passend zum zu prüfenden Datensatz gewählt werden. Das Prüfergebnis ist bei allen Checks der Gruppe B eine einfache Aussage, ob der Test bestanden wurde, woraus sich ein Hinweis auf die Plausibilität bzw. Konsistenz der Modellierung ergibt, der vom Nutzer zu bewerten ist.

3.2 Definition des Prüfplans

Die in Kapitel 3.1 beschriebenen Qualitätselemente sind wesentlicher Bestandteil des Prüfplans. Dieser soll eindeutig, nachvollziehbar und automatisiert zu verarbeiten sein. Ein XML-basiertes Format erfüllt diese Voraussetzungen. Alle notwendigen Elemente können entsprechend dem XML-Schema in Anhang A dargestellt werden.

Der Prüfplan besteht im Wesentlichen aus einer Liste von Checks, die gemäß den Modellanforderungen durch Parameter näher definiert werden. Checks und Parameter werden nicht starr vorgegeben, sondern in Form einer komplexen Datenstruktur definiert, die alle notwendigen Subelemente sowie deren Datentypen enthält.

3.3 Spezifikation am Beispiel des Modellierungshandbuches der SIG3D

Im Rahmen der Neuerstellung des Stadtmodells Ludwigsburg wurden die Empfehlungen des Modellierungshandbuchs der SIG3D (SIG-3D Quality Working Group, 2012) zur Beschreibung der Anforderungen genutzt. Folgender Katalog von Anforderungen lag der Modellierung zu Grunde:

1. LOD2 Modell nach CityGML 2.0
2. die 3D-Geometrie wird auf den Gebäudegrundrissen des Katasters aufgebaut
3. *Solid*-Geometrie referenziert *BoundarySurfaces*
4. nur *Buildings*, da keine klaren Kriterien zur Abgrenzung von *BuildingParts*
5. *GroundSurface* horizontal auf Höhe des niedrigsten Punktes des Gebäudegrundrisses
6. Keine *BuildingInstallations*
7. Building-Attribute: *ID* (nach Kataster), *creationDate*, *function*, *roofType*, *measuredHeight* (berechnet), sowie die generischen Attribute *Min Hoehe Trauflinie*, *Min Gelaendepunkt*, *Max Hoehe Firstlinie*, *Datenquelle Dachhoehe*, *StandLK*, *Methode*, *Datenquelle Bodenhoehe*, *Datenquelle Lage*

8. Keine Dachüberstände
9. Keine *OuterFloorSurface*, *OuterCeilingSurface*, *ClosureSurface* Elemente

Das Beispiel zeigt, dass neben der Geometrie und einigen semantischen Entitäten auch alle Attribute spezifiziert wurden. Manche Attribute erlauben eine formale Plausibilitätsprüfung, z.B. die Eindeutigkeit der vorkommenden Gebäude-IDs oder die Kohärenz von Werten und Codelisten. Sämtliche Vorgaben können im Prüfplan festgelegt werden. Wenn eine Implementierung der zugehörigen Checks vorhanden ist, lässt sich der Prüfplan vollständig automatisch verarbeiten. Diese Einschränkung ist wichtig, da nicht alle verbal formulierten Anforderungen in einem formalen Prüfplan dargestellt werden können. Im obigen Beispiel trifft dies auf die Punkte 2 und 5 zu. Ein Auszug aus dem zugehörigen XML-Instanzdokument des Prüfplans ist in Anhang C dargestellt.

4 Prüfbericht

Das Ergebnis der Prüfung ist der Prüfbericht. Neben einem Überblick über die Modelleigenschaften enthält er alle bei der Validierung gegen den Prüfplan festgestellten Abweichungen, die im geprüften Datensatz auftreten. Er wird daher auch als Fehlerprotokoll oder Error Report bezeichnet. Seine Struktur ergibt sich aus dem Prüfplan. Mit jedem Qualitätselement des Prüfplans korrespondiert ein Fehlerelement gleichen Typs im Prüfbericht. Damit kann für jeden gefundenen Fehler eine Instanz des jeweiligen Typs erzeugt werden, die ihn vollständig beschreibt. Die Beschreibung umfasst die ID des betroffenen Elements, die Art und Ausprägung des Fehlers.

Listing 2: Beispiel zum Reporting eines Planaritätsfehlers

```
<cd:buildingRecord ID="DEBW_LOD2_1013525">
  <cd:numberOfPolygonsLod1>0</cd:numberOfPolygonsLod1>
  <cd:numberOfPolygonsLod2>7</cd:numberOfPolygonsLod2>
  <cd:numberOfGroundsurface>1</cd:numberOfGroundsurface>
  <cd:numberOfWallsurface>4</cd:numberOfWallsurface>
  <cd:numberOfRoofsurface>2</cd:numberOfRoofsurface>
  <cd:numberOfUnclassifiedsurface>0</cd:numberOfUnclassifiedsurface>
  <cd:numberOfEdges>15</cd:numberOfEdges>
  <cd:errorList>
    <cd:error>
      <cd:checkIdentifier>CP_PLAN</cd:checkIdentifier>
      <cd:specification>
        <cd:featureType>polygon</cd:featureType>
        <cd:featureId>6</cd:featureId>
        <cd:value>0.0316</cd:value>
      </cd:specification>
      <cd:errorDetectionTime>2014-01-31T12:00:00</cd:errorDetectionTime>
    </cd:error>
  </cd:errorList>
</cd:buildingRecord>
```

Zusätzlich können noch generelle Modelleigenschaften und statistische Werte in den Bericht integriert werden. Damit lassen sich allgemeine Aussagen über die Struktur des Stadtmodells treffen sowie über dessen Eignung gemäß der Prüfanforderungen. Als Maßzahlen für die Güte eines Stadtmodells in Bezug zu den Prüfkriterien können neben dem Anteil der fehlerfreien

Gebäude die aggregierten Werte der jeweiligen Fehlertypen dienen. Das XML-Schema für den Prüfbericht ist in Anhang B dokumentiert.

Die aktuelle Implementierung des CityDoctor Validierungs-Tools speichert die Prüfergebnisse extern in einer XML-Datei und verwendet dazu Fehlerelemente gemäß Anhang B. Der geprüfte Datensatz wird dabei nicht verändert. Die Zuordnung gefundener Fehler erfolgt auf Gebäudeebene über den Identifikator des *Building*-Elements. Neben der reinen Fehlerinformation sind statistische Daten zur geometrisch-semantischen Struktur des Gebäudes enthalten. Jedes Gebäude kann beliebig viele *error*-Elemente enthalten. Listing 2 zeigt ein Beispiel für einen Planaritätsfehler in Polygon 6 mit einer maximalen Abweichung von gut 31 mm als Ergebnis der Prüfung CP_PLAN. Die ID des betroffenen Gebäudes sowie eine Zusammenfassung seiner wichtigsten Bestandteile werden mit angegeben.

5 Signatur des Prüfberichts

Um den Aufwand einer vollständigen Qualitätsprüfung nicht bei jeder Datenweitergabe durchführen zu müssen, ist die Signierung des Prüfberichts möglich. Damit kann der Bericht dem geprüften Datensatz eindeutig zugeordnet und eine Manipulation nach der Prüfung weitgehend ausgeschlossen werden. Das CityDoctor Tool implementiert diese Möglichkeit exemplarisch.

Dazu wird ein SHA-1-Hashwert (National Institute of Standards and Technology, 2012) der geprüften Datei berechnet und im Prüfbericht gespeichert. Falls eine höhere Manipulationssicherheit erforderlich wird, lässt sich dies durch die Verwendung eines stärkeren Verfahrens zur Hasherzeugung erreichen (Fox, 2013). Eine Signierung ist derzeit für CityGML-Dateien implementiert. Kommen die zu prüfenden Daten aus einer Geodatenbank oder von einem Geodatenserver, ist als Zwischenschritt der Export in eine Datei erforderlich.

6 Zusammenfassung und Ausblick

Die Spezifizierung eines XML-basierten Formats zur Definition von Prüfplänen und Prüfergebnissen ermöglicht die explizite Dokumentation der Qualität von 3D-Stadtmodellen. Sie können als definierte Schnittstelle genutzt werden, wodurch die Unabhängigkeit des Prüfergebnisses vom verwendeten Tool gegeben ist. Wie sich gezeigt hat, lassen sich Anforderungen an ein Stadtmodell nicht in allen Fällen in einen Prüfplan umsetzen bzw. nur schwer formalisiert erfassen. Durch weitere Untersuchungen sollte geklärt werden, wo die Grenzen der Ausdrucksmächtigkeit eines formalen Prüfplans liegen.

Zusätzlich enthält das Prüfergebnis alle notwendigen Informationen für eine eventuell erforderliche Heilung des Modells, die weitgehend automatisiert angeschlossen werden kann. Auch hier wird die Schnittstelleneigenschaft des formalen Prüfberichts nutzbar, weshalb die Heilung unabhängig vom Prüftool durchgeführt werden kann. Für eine vollständige Unabhängigkeit des Heilungsprozesses von der Prüfung ist allerdings noch eine allgemein anerkannte Definition der Fehlertypen erforderlich. Die bisherigen Prüfergebnisse der beiden Tools CityDoctor und val3dity sind u.a. hinsichtlich des verwendeten Geometriemodells und der erzeugten Fehlerobjekte nicht direkt übertragbar. Im Rahmen eines Interoperability-Experiments

der OGC soll untersucht werden, wie verschiedene Ansätze bei der Prüfung miteinander korrespondieren und inwieweit die Ergebnisse übertragbar sind. Der Beitrag ist auch in diesem Zusammenhang als Diskussionsgrundlage auf dem Weg zu einer Vereinheitlichung der Qualitätsprüfung im Bereich 3D-Stadtmodelle zu verstehen.

7 Literaturverzeichnis

- CZERWINSKI, A., PLÜMER, L., 2008. Landesweite Web Services der GDI NRW und CityGML - erfolgreicher Einsatz für die EU-Umgebungslärmkartierung – GIS.Business 7, 41–43.
- FOX, D., 2013. Secure Hash Algorithm SHA-3. Datenschutz Datensicherheit 37, 104.
- GRÖGER, G., COORS, V., 2010. Rules for validating GML geometries in CityGML.
http://files.sig3d.de/file/ag-qualitaet/20110531_Regeln_GML_final_EN.pdf (abgerufen 31.1.2014)
- KADEN, R., KRÜGER, A., KOLBE, T.H., 2012. Integratives Entscheidungswerkzeug für die ganzheitliche Planung in Städten auf der Basis von semantischen 3D-Stadtmodellen am Beispiel des Energieatlases Berlin. In: Tagungsband der Wissenschaftlich-Technischen Jahrestagung der DGPF 2012, Potsdam.
- KUNZE, C., HECHT, R., HAHMANN, S., 2013. Zur Vollständigkeit des Gebäudedatenbestandes von OpenStreetMap. Kartogr Nachr 63, 73–81.
- SIG-3D QUALITY WORKING GROUP, 2012. Handbuch für die Modellierung von 3D Objekten - Teil 2: Modellierung Gebäude (LOD1, LOD2 und LOD3).
[http://wiki.quality.sig3d.org/index.php/Handbuch_f%C3%BCr_die_Modellierung_von_3D_Objekten_-_Teil_2:_Modellierung_Geb%C3%A4ude_\(LOD1,_LOD2_und_LOD3\)](http://wiki.quality.sig3d.org/index.php/Handbuch_f%C3%BCr_die_Modellierung_von_3D_Objekten_-_Teil_2:_Modellierung_Geb%C3%A4ude_(LOD1,_LOD2_und_LOD3)) (abgerufen 31.1.2014)
- STADLER, A., KOLBE, T.H., 2007. Spatio-semantic Coherence in the Integration of 3D City Models. In: Proceedings of the 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede.
- WAGNER, D., ALAM, M.N., COORS, V., 2013a. Geometric validation of 3D city models based on standardized quality criteria. In: Urban and Regional Data Management: UDMS Annual 2013. Taylor and Francis Group, London, pp. 197–210.
- WAGNER, D., BOGDAHN, J., WEWETZER, M., ALAM, M.N., PRIES, M., COORS, V., 2013b. Geometric-semantical consistency validation of CityGML models. In: Pouliot, J., Daniel, S., Hubert, F., Zamyadi, A. (Eds.), Progress and New Trends in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg.
- WEWETZER, M., FALKENHAUSEN, J., WAGNER, D., ALAM, M.N., PRIES, M., COORS, V., FISCHER, J., 2013. Verbundprojekt CityDoctor - Entwicklung von Methoden und Metriken zum Qualitätsmanagement virtueller Stadtmodelle. In: Forschungsbericht 2012 - Angewandte Forschung Zur Stadt Der Zukunft. Logos Verlag, Berlin, pp. 15–21.
- ZHAO, J., STOTER, J., LEDOUX, H., 2014. A framework for the automatic geometric repair of CityGML models. In: Buchroithner, M., Prechtel, N., Burghardt, D. (eds.), Cartography from Pole to Pole - Selected Contributions to the XXVIth International Conference of the ICA, Dresden 2013. Springer Berlin Heidelberg, pp. 187–202.
- ZIELSTRA, D., ZIPF, A., 2010. Quantitative studies on the data quality of OpenStreetMap in Germany, in: GIScience. Sixth International Conference on Geographic Information Science, Zürich, pp. 20–26.

Anhang A: Prüfplan (XML-Schema)

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://citydoctor.hft-stuttgart.de" xmlns:cd="http://citydoctor.hft-
stuttgart.de"
  elementFormDefault="qualified">

  <xs:element name="checks">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cd:check" maxOccurs="unbounded"/></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="checkId" type="xs:string"/></xs:element>
  <xs:attribute name="id" type="xs:string"/></xs:attribute>

  <xs:element name="check">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cd:checkId"/></xs:element>
        <xs:element ref="cd:parameter" minOccurs="0"
          maxOccurs="unbounded">
        </xs:element>
        <xs:element ref="cd:prerequisites" minOccurs="0"
          maxOccurs="unbounded">
        </xs:element>
        <xs:element ref="cd:active"/></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="parameter">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cd:parameterName"/></xs:element>
        <xs:element ref="cd:parameterValue"/></xs:element>
        <xs:element ref="cd:parameterOfElement" minOccurs="0">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="prerequisites">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cd:checkId" maxOccurs="unbounded"/></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="ValidationPlan">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cd:checks"/></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="parameterName" type="xs:string"/></xs:element>
  <xs:element name="parameterValue" type="xs:string"/></xs:element>
  <xs:element name="parameterOfElement" type="xs:string"/></xs:element>
  <xs:element name="active" type="xs:boolean"/></xs:element>
</xs:schema>

```

Anhang B: Prüfbericht/Fehlerprotokoll (XML-Schema)

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://citydoctor.hft-
stuttgart.de" xmlns:cd="http://citydoctor.hft-stuttgart.de" elementFormDefault="qualified">

  <xs:element name="validationReport" type="cd:validationReportType"/></xs:element>
  <xs:element name="buildingRecord" type="cd:buildingRecordType"/></xs:element>
  <xs:element name="errorList" type="cd:errorListType"/></xs:element>
  <xs:element name="error" type="cd:errorType"/></xs:element>
  <xs:element name="errorDetectionTime" type="xs:dateTime"/></xs:element>
  <xs:element name="specification" type="cd:specificationType"/></xs:element>

  <xs:complexType name="validationReportType">
    <xs:sequence>
      <xs:element name="dataSetName" type="xs:string"/></xs:element>
      <xs:element name="dataSetHash" type="xs:hexBinary"/></xs:element>
      <xs:element name="dataSetProperties" type="cd:dataSetPropertiesType"/></xs:element>
      <xs:element name="validationProperties" type="cd:validationPropertiesType"/></xs:element>
      <xs:element name="validationTime" type="xs:dateTime"/></xs:element>
      <xs:element name="buildingRecord" type="cd:buildingRecordType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="validationPropertiesType">
    <xs:sequence>
      <xs:element name="selectedChecks">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="cd:checkIdentifier"/></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="dataSetPropertiesType">
    <xs:sequence>
      <xs:element name="numberOfBuildings" type="xs:int"/></xs:element>
      <xs:element name="numberOfBuildingParts" type="xs:int"/></xs:element>
      <xs:element name="lod1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="buildingsLod1SolidGeometry" type="xs:int"/></xs:element>
            <xs:element name="buildingsLod1MultiSurfaceGeometry" type="xs:int"/></xs:element>
            <xs:element name="numberOfBuildingPartsLod1" type="xs:int"/></xs:element>
            <xs:element name="numberOfPolygons" type="xs:int"/></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="lod2">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="buildingsLod2SolidGeometry" type="xs:int"/></xs:element>
            <xs:element name="buildingsLod2MultiSurfaceGeometry" type="xs:int"/></xs:element>
            <xs:element name="numberOfBuildingPartsLod2" type="xs:int"/></xs:element>
            <xs:element name="numberOfPolygons" type="xs:int"/></xs:element>
            <xs:element name="numberOfGroundSurfaces" type="xs:int"/></xs:element>
            <xs:element name="numberOfWallSurfaces" type="xs:int"/></xs:element>
            <xs:element name="numberOfRoofSurfaces" type="xs:int"/></xs:element>
            <xs:element name="numberOfOtherSurfaceTypes" type="xs:int"/></xs:element>
            <xs:element name="numberOfUnclassifiedPolygons" type="xs:int"/></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

<xs:complexType name="buildingRecordType">
  <xs:sequence>
    <xs:element name="numberOfPolygonsLod1" type="xs:int"/></xs:element>
    <xs:element name="numberOfPolygonsLod2" type="xs:int"/></xs:element>
    <xs:element name="numberOfGroundsurface" type="xs:int"/></xs:element>
    <xs:element name="numberOfWallsurface" type="xs:int"/></xs:element>
    <xs:element name="numberOfRoofsurface" type="xs:int"/></xs:element>
    <xs:element name="numberOfUnclassifiedsurface" type="xs:int"/></xs:element>
    <xs:element name="numberOfEdges" type="xs:int"/></xs:element>
    <xs:element ref="cd:errorList"/></xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string"/></xs:attribute>
</xs:complexType>

<xs:complexType name="errorListType">
  <xs:sequence>
    <xs:element ref="cd:error" maxOccurs="unbounded"/></xs:element></xs:sequence>
</xs:complexType>

<xs:complexType name="errorType">
  <xs:sequence>
    <xs:element ref="cd:checkIdentifier"/></xs:element>
    <xs:element ref="cd:specification"/></xs:element>
    <xs:element ref="cd:errorDetectionTime" minOccurs="0"/></xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="specificationType">
  <xs:sequence>
    <xs:element name="featureType" type="xs:string"/></xs:element>
    <xs:element name="featureId" type="xs:string" minOccurs="0"/></xs:element>
    <xs:element name="value" type="xs:string" minOccurs="0"/></xs:element>
  </xs:sequence>
</xs:complexType>

<xs:element name="checkIdentifier">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="CP_NUMPOINTS"/></xs:enumeration>
      <xs:enumeration value="CP_CLOSE"/></xs:enumeration>
      <xs:enumeration value="CP_DUPPOINT"/></xs:enumeration>
      <xs:enumeration value="CP_NULLAREA"/></xs:enumeration>
      <xs:enumeration value="CP_SELFINT"/></xs:enumeration>
      <xs:enumeration value="CP_PLANNATIVE"/></xs:enumeration>
      <xs:enumeration value="CS_NUMFACES"/></xs:enumeration>
      <xs:enumeration value="CS_SELFINTNATIVE"/></xs:enumeration>
      <xs:enumeration value="CS_OUTEREDGE"/></xs:enumeration>
      <xs:enumeration value="CS_OVERUSEDEDGE"/></xs:enumeration>
      <xs:enumeration value="CS_FACEORIENT"/></xs:enumeration>
      <xs:enumeration value="CS_FACEOUT"/></xs:enumeration>
      <xs:enumeration value="CS_UMBRELLA"/></xs:enumeration>
      <xs:enumeration value="CS_CONCOMP"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD2_ROOF"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD2_WALL"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD2_GROUND"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD1_ASSOLID"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD1_BUILDPARTS1"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD1_BUILDPARTS2"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD1_BUILDPARTS3"/></xs:enumeration>
      <xs:enumeration value="SEM_LOD1_NUMFLOORS"/></xs:enumeration>
      <xs:enumeration value="C_MSIFSOLID"/></xs:enumeration>
      <xs:enumeration value="C_BNBPIFSOLID"/></xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:schema>

```

Anhang C: Instanzdokument eines Prüfplans (Auszug)

```

<?xml version="1.0" encoding="UTF-8"?>
<cd:ValidationPlan xmlns:cd="http://citydoctor.hft-stuttgart.de"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://citydoctor.hft-
stuttgart.de ValidationPlan.xsd ">

  <cd:checks>
<!-- geometry checks -->
  ...
    <cd:check>
      <cd:checkId>CS_FACEOUT</cd:checkId>
      <cd:prerequisites>
        <cd:checkId>CS_FACEORIENT</cd:checkId>
      </cd:prerequisites>
    </cd:check>
    <cd:check>
      <cd:checkId>CS_NUMFACES</cd:checkId>
    </cd:check>
    <cd:check>
      <cd:checkId>CS_SELFINT</cd:checkId>
      <cd:parameter>
        <cd:parameterName>geometricTolerance</cd:parameterName>
        <cd:parameterValue>0.01</cd:parameterValue>
      </cd:parameter>
      <cd:prerequisites>
        <cd:checkId>CP_PLAN</cd:checkId>
        <cd:checkId>CS_NUMFACES</cd:checkId>
      </cd:prerequisites>
    </cd:check>
  ...

<!-- structure/type of geometry, semantic consistency -->
  <cd:check>
    <cd:checkId>C_MSIFSOLID</cd:checkId>
  </cd:check>
  <cd:check>
    <cd:checkId>SEM_MEASURED_HEIGHT</cd:checkId>
    <cd:parameter>
      <cd:parameterName>geometricTolerance</cd:parameterName>
      <cd:parameterValue>0.1</cd:parameterValue>
    </cd:parameter>
  </cd:check>

<!-- attribute checks -->
  <cd:check>
    <cd:checkId>SEM_MANDATORY_ATTRS</cd:checkId>
    <cd:parameter>
      <cd:parameterName>mandatoryAttribute</cd:parameterName>
      <cd:parameterValue>gml_id</cd:parameterValue>
      <cd:parameterOfElement>Building</cd:parameterOfElement>
    </cd:parameter>
    <cd:parameter>
      <cd:parameterName>mandatoryAttribute</cd:parameterName>
      <cd:parameterValue>function</cd:parameterValue>
      <cd:parameterOfElement>Building</cd:parameterOfElement>
    </cd:parameter>
  ...
  </cd:check>
  <cd:check>
    <cd:checkId>SEM_ATTRIBUTE_DOMAIN</cd:checkId>
    <cd:parameter>
      <cd:parameterName>function</cd:parameterName>
      <cd:parameterValue>./[pathToCodelist]</cd:parameterValue>
    </cd:parameter>
  ...
  </cd:check>
</cd:checks>
</cd:ValidationPlan>

```