Versioning of 3D City Models for Municipality Applications: Needs, Obstacles and Recommendations

Helen Eriksson1,2,* and Lars Harrie2

1 Lantmäteriet, Österleden 15, SE-981 38 Kiruna, Sweden
2 Department of Physical Geography and Ecosystem Science, Lund University, Sölvegatan 12, SE-223 62 Lund, Sweden; lars.harrie@nateko.lu.se
* Correspondence: helen.eriksson@lm.se

Abstract: The use of 3D city models is changing from visualization to complex use cases where they act as 3D base maps. This requires links to registers and continuous updating of the city models. Still, most models never change or are recreated instead of updated. This study identifies obstacles to version management of 3D city models and proposes recommendations to overcome them, with a main focus on the municipality perspective, foremost in the planning and building processes. As part of this study, we investigate whether national building registers can control the version management of 3D city models. A case study based on investigations of standards, interviews and a review of tools is presented. The study uses an architectural model divided into four layers: data collection, building theme, city model and application. All layers require changes when implementing a new versioning method: the data collection layer requires restructuring of technical solutions and work processes, storage of the national building register requires restructuring, versioning capabilities must be propagated to the city model layer, and tools at the application layer must handle temporal information better. Strong incentives for including versioning in 3D city models are essential, as substantial investment is required to implement versioning in all the layers. Only capabilities required by applications should be implemented, as the complexity grows with the number of versioning functionalities. One outcome of the study is a recommendation to link 3D city models more closely to building registers. This enables more complex use in, e.g., building permits and 3D cadastres, and authorities can fetch required (versioning) information directly from the city model layer.

Keywords: version management; temporal information; 3D city models; 3D buildings

1. Introduction

Cities around the world are developing 3D city models, and the number is growing [1,2]. These models are mainly used for visualization, but they are increasingly also used for analyses, urban planning and as 3D base maps. The latter, more advanced, use of 3D city models requires links to other registers and continuous updating of the city models.

Versioning capabilities are included in a number of geodata standards and specifications, such as CityGML 2.0 [3], the proposed version 3.0 of CityGML [4,5], the Infrastructure for spatial information in Europe (INSPIRE) Building specification [6], and the proposed modified Git version control system implemented in CityJSON [7]. The ISO standard Product Lifecycle Support (PLCS) [8] is another example; it has been used for the lifecycle management of building information modelling (BIM) data [9], and has the potential to also improve the versioning of 3D city models [10]. Even though many versioning methods exist today, most 3D city models have no version management, and many of them are not changed at all. Three-dimensional city models that are changed are often recreated instead of updated [7].

Several applications require temporal information and versioning of 3D city models. Of special interest in this study is the municipal perspective on applications in the planning and building sector. This includes, among others, the potential to create legal documents...
based on 3D city models. Examples of this are: (1) detailed development plans for urban planning—these plans are still often shared in paper format or as pdf files [11]; (2) building permit decisions—also, here the information is often shared in paper format or as pdf files [12,13] which can make the process ineffective. Much would be gained if the work process had a more standardized and process-oriented approach where all actors can share information [14,15]; and (3) 3D cadastral dossiers, where the 3D cadastre still is defined as 2D drawings with textual descriptions [16]. Here, both BIM models and 3D geodata building models could improve the process [17–19]. Common to all of these examples is that it is important to save and archive not only the legal documents, but also the background material that was used to arrive at the decision. Furthermore, these examples require that the 3D city models have connections to the national building register.

The overall aim of this study is to identify obstacles to the version management of 3D city models and propose recommendations on how to overcome these obstacles, mainly focusing on the municipality perspective on applications in the planning and building sector. One specific aim is to investigate whether a national building register can be used to control the version management of 3D city models, and, if so, which requirements must be set on the specification for a national building register.

To identify obstacles, we evaluate the data flow between layers, from data collection to the end user, focusing on the version management of building information (Figure 1). The building theme layer in Figure 1 represents the national storage of building information. Here, the information is stored according to national specifications, for example, in a national building register or as BIM data. Updates to the building information (from geodetic surveys, the building permit process, etc.) are made to this layer and the data collection is mainly performed at the local level, for example, at a municipality. The city model layer represents the distribution of 3D city model information from a national platform. It includes a 3D city model with information on several themes (e.g., buildings, roads and bridges) in standardized formats according to, for example, national profiles of the CityGML standard or the CityJSON specification. Finally, tools in the application layer use the 3D city model information for visualization, analysis, urban planning, etc.

**Figure 1.** Overview of an architectural model for the provision of 3D city model information.
To enable applications to utilize data versioning, in this case for building information, proper temporal information must be collected and propagated through all the layers. This requires version management for datasets at all layers and mapping between these version management systems. Therefore, to identify obstacles to the versioning of 3D city models, we evaluate the following issues:

1. Data provider incentives: Do the data providers, mainly municipalities, have enough incentives to version their 3D city models?
2. Collection of versioning information: Are there technical or organizational issues when collecting the lifecycle information for buildings?
3. Database implementation on building theme layer: Are the versioning methods too difficult to implement?
4. Capability of versioning management in the building register: Do versioning methods in the building register fulfil versioning requirements of applications in the planning and building sector?
5. Mapping versioning information between layers: Can versioning information from one layer be propagated to another layer without information loss (in the direction of the arrows in Figure 1)?
6. Application tools: Are the application tools for 3D city models sufficient for handling version management data?

There are many aspects on why so few 3D city models are versioned, for example, legal aspects, business reasons, security and data privacy issues, competition between actors (e.g., between software vendors) and lack of technical competence in, for example, the municipalities. In this study, we focus on the technical aspects of version management.

The next section is a literature review, including descriptions of why temporal information and version management are needed in 3D city models and of versioning methods in 3D city model standards, and their practical use. Section 3 describes case studies of version management of 3D city models based on the six issues described above. Issues 2 to 5 are exemplified by the way 3D city models are managed in Sweden, but conditions are similar in many countries, which makes the case study also interesting to a larger community. The case study includes interviews with personnel in Swedish municipalities, an investigation of version management in geodata standards, and an evaluation of software tools. The paper ends with a recommended approach to selecting versioning methods and conclusions.

2. Literature Review

2.1. Need for Temporal Information and Version Management in 3D City Models

Chaturvedi and Kolbe [20] described the need for 3D city models to support time-dependent properties. They distinguish between quantitative changes, describing functions of time on an object, e.g., energy consumption for a building varying over time, and qualitative changes, where objects (e.g., buildings) are created and disappear over time. Seven requirements on how 3D city models should be extended to handle both of these types are defined: (1) linking sensors and the Internet of Things (IoT) with city objects, (2) providing events and alerts, (3) moving objects, (4) supporting time series in-line within cities, (5) supporting complex patterns and schedules, (6) managing historic versions. How 3D city models should be extended with additional time-dependent properties (concerning geometry, theme, topology and appearance) to comply with these requirements was also described. Three 3D city models (Industry Foundation Classes (IFC) 4, INSPIRE and CityGML 2.0) were evaluated against these requirements, and the results show that all models have complete or limited support for some requirements, but no models support all. Three CityGML Application Domain Extensions (ADEs) were also tested, and these had complete support of requirements within their fields. Finally, this was compared to functionalities in the proposed CityGML 3.0. Here, two of the tested ADEs, Dynamizer and Versioning, are included as new modules, extending the support in CityGML in these areas.
Biljecki et al. [1] performed a literature review on how 3D city models are used in different domains. It describes a wide use of 3D city models and categorizes usage into 29 use case groups, such as urban planning, routing, 3D cadastre and emergency response. Analyzing their results from this review from a temporal viewpoint, only three use cases clearly mention the temporal dimension: planning, change detection and archaeology. Fourteen use cases perform estimations, predictions or simulations, which indicates that several simultaneous versions of the 3D city model could be needed and indirectly indicates the need for version management. Examples of such use cases are estimation of solar irradiation and forecasting seismic damage.

Another application that requires temporal information is 3D cadastre. Sun et al. [19] described a technical and legal solution by which construction companies and surveying units share information for handling 3D cadastre information. Their method was based on open standards: cadastre information was stored in the Land Administration Domain Model (LADM) [21], and the physical extents of cadastre units were stored in the ISO standard Industry Foundation Classes (IFC) [22]. The IFC data (building and cadastre borders) were converted to CityGML 3.0 and integrated into an existing 3D city model. This cadastre information can be visualized on a city scale and be used for macro analysis by linking CityGML 3.0 data to LADM. Temporal information was not included, but developing a collaborative workflow among actors in a lifecycle project was mentioned as a future requirement.

2.2. Temporal Modelling of Geodata

Temporal information in spatiotemporal information systems can be organized in several ways. Worboys and Duckham [23] defined four stages: Static representation: single moments in time; The snapshot metaphor: temporal snapshots describing a state at a certain time; Object lifelines: changes of state over time for objects, attributes and their relations; and Events, actions and processes: continuants (things that endure) and occurrents (things that happens and then are gone). Worboys [24] described another view, in which a geographic phenomenon is described in an event-oriented way. Here, time is defined as a collection of tick events ordered as sets of channels, referenced to as a clock.

Geodata that include temporal information can be stored in spatiotemporal databases, where changes over time in the geometry, attributes and topology of objects are captured and can be queried. Two types of time representations are often used: transaction time, when changes occur to objects in the database, and valid time, which reflects the time when things happen in reality (e.g., a building is destroyed). Datasets can include both transaction time and valid time, which is denoted bi-temporal modelling. These timestamps can be set on different levels. Taking buildings as an example, the timestamp could be on the entire building, on building parts, on the geometry or on individual attributes. The more detailed levels that are timestamped, the more storage capacity will be required, but temporal changes can also be traced at finer granularity [25].

2.3. Versioning Methods for 3D Buildings and 3D City Models

This section describes capabilities of version management systems for 3D city models that vary from simple to more comprehensive. However, it is definitely not always the case that comprehensive systems are the best; it all depends on the application requirements. Using a more complicated method than necessary entails unnecessary complexity in data handling. This topic is evaluated and discussed in detail in [10], where a comparison study of version management in PLCS, CityGML 3.0 and the modified Git proposal in the context of 3D cadastre was performed.

2.3.1. Versioning in CityGML Versions 2.0 and 3.0

The CityGML standard by the Open Geospatial Consortium (OGC) is the most comprehensive standard today for the exchange of 3D geodata city models [3,26]. Its main focus is to represent the geometric and semantic aspects of features in a city. The current version,
CityGML 2.0, is being revised to version 3.0 [5,27] both to increase usage of the standard in different applications and to improve interoperability with other relevant standards.

The temporal information in CityGML 2.0 includes the attributes creationDate–terminationDate, representing transaction time, and the attributes yearOfConstruction and yearOfDemolition of a building. CityGML 3.0 also includes creationDate–terminationDate and added the attribute dateOfRenovation (based on INSPIRE Building, see Section 2.3.2). In CityGML 2.0, these date attributes belong to the AbstractBuilding feature, and in CityGML 3.0 to the AbstractConstruction feature, that is, features that both Building and BuildingPart inherit from, and, consequently, the attributes are included in both buildings and building parts. CityGML 3.0 further includes the attributes validTo–validFrom, referring to valid time, and a new versioning module containing the feature types Version, VersionTransition and Transaction. This makes it possible to describe transactions and define which objects belong to a certain version. Versions are also allowed to fork in order to, e.g., represent alternative plans of a city [4].

2.3.2. Versioning in INSPIRE Building

INSPIRE Building is one of 34 themes in the INSPIRE Directive [28], and the data model in its data specification has been strongly influenced by CityGML 2.0, and has also in turn influenced CityGML 3.0. Some attributes are commonly used in all INSPIRE themes, such as attributes for transaction time, beginLifespanVersion and endLifespanVersion. In the building theme, additional attributes specifying certain stages are also included: dateOfConstruction, dateOfRenovation and dateOfDemolition, together with a versioning identifier, versionId.

2.3.3. Modified Git Versioning Proposal Implemented in CityJSON

A modified Git versioning method was proposed by Vitalis et al. [7]. It has a data structure similar to the directed acyclic graph (DAG) structure in Git, a popular version control system mainly for programming, but also for work coordination and project management. The DAG consists of nodes representing objects or data, directed edges representing relationships between nodes, a root node and leaf nodes. All nodes know their parent, so the graph can be traversed from leaves to root. Vitalis et al. [7] implemented their proposal in CityJSON, a JavaScript Object Notation (JSON) implementation of the major part of the CityGML 2.0 data model [29]. City objects are listed as CityObjects properties and versions as versioning properties. A version includes building objects belonging to this version, an author property, a date property for date of a version, a parents property linking to the previous version, and a message property.

2.3.4. Spatio-Semantic Comparison of CityGML Models

It can be difficult to detect spatiotemporal changes in a CityGML model due to its complex hierarchical structures, syntactic and geometric ambiguities, and the combination of graphical and semantic aspects of city objects in one place. Therefore, CityGML models are often replaced instead of updated, and thus the track of changes is lost. To overcome this, Nguyen et al. [30] proposed using graph databases to detect changes on the semantic and geometric levels of CityGML datasets. They implemented their graph database in neo4j (https://neo4j.com.neo4j-graph-database/), a native open-source NoSQL graph database. An old and a new CityGML dataset were mapped to the graph database, and thereafter the node properties and geometries of the two resulting graphs were matched. All changes made to the old CityGML dataset in the test case were found, and it was updated accordingly using a Web Feature Service (WFS).

Nguyen and Kolbe [31] extended the above graph database proposal to also include user-oriented interpretation of detected changes, as different users and stakeholders can have different interests in changes of 3D city models. They proposed an enhanced mapping and matching process where detected changes are divided into different edit operations and the changes are categorized based on semantic content (Figure 2).
Figure 2. CityGML datasets are mapped to the graph database and graphs are matched. Different edit operations are detected, and changes are categorized based on their semantic content (from Nguyen and Kolbe [31]).

The relevance of the categories of changes is further determined with respect to requirements and expectations by users and stakeholders. While developers are interested in all changes, city administrators are mainly interested in top-level changes (e.g., changes to a building), and data administrators are interested in thematic, geometric, structural and top-level changes.

2.4. Current Use of Versioning Methods for 3D City Models

Many applications that use 3D city models require that the models are up to date, that it is possible to retrieve information about city objects from previous moments in time, and that alternative versions of a city model can be managed simultaneously. This is reflected in the addition of a new versioning module to the proposed version 3.0 of CityGML [4]. Still, a majority of openly available 3D city models are never changed, and the models that are changed are recreated instead of updated [7,30]. Having 3D city models that include temporal information and version management is a relatively new phenomenon, and we were unable to find research papers about implementation of 3D city model versioning or issues that could hamper such implementation. One glimpse was found in a Dutch study [32], where interviews were conducted in six cities about the content of their 3D city models, and one topic concerned the actuality of their models. The models originated from 2000 to 2019. Three cities did full updates of the model, one on a yearly basis. The other cities updated changed parts only, one on a daily basis and the others periodically.

Many cities that use CityGML for 3D city models use a 3DCityDB database (https://www.3dcitydb.org) to store and manage their models. 3DCityDB is an open-source project that provides free database schema implementations to be used on top of standard spatial relational databases. Recently, links have also been established between 3DCityDB and BIM data [33]. Examples of where 3DCityDB is used include Germany, where federal states submit 3D building information to a central 3DCityDB database with around 40 million building objects [34], and Vienna, where a 3D city model is stored in 3DCityDB for computation of energy demands in buildings and for visualization [35,36]. In Sweden, several municipalities are also using, or are planning to use, 3DCityDB.

Versioning capability is currently not included in 3DCityDB, but a slightly modified version of 3DCityDB was used by Doulamis et al. [37] to develop a 4D modelling framework (3D and time) for a land information management system that visualizes a 3D city model of an area over time. Doulamis et al. [37] also mentioned that the proposed CityGML 3.0 includes temporal objects that, if implemented, could improve the time dimension and open up for better versioning of 3D city models in 3DCityDB in the future.
3. Case Study: Issues for Versioning of Building Information

The case study focuses on the six version management issues stated in the introduction: (1) data provider incentives, (2) collection of versioning information, (3) database implementation on building theme layer, (4) capability of versioning management in the building register, (5) mapping of versioning information between layers, and (6) application tools for 3D city models. Issues 2 to 5 have been studied in the context of 3D city models for Swedish municipalities, with a special focus on building information. The background to the current 2020 and proposed future handling of building information in Sweden is provided in Section 3.1. In Section 3.2, the information gathering for the case study is described, and Sections 3.3–3.8 provide detailed descriptions of the six issues, including discussions.

3.1. Background

A main focus of this case study is if and how the national building register could be used in the updating and versioning process of 3D city models. In Sweden, the national building register (building theme layer in Figure 1) is managed in collaboration between the municipalities and the national mapping authority (NMA), Lantmäteriet. Buildings that are affected by property formation or building permits must be updated by the municipality according to the current building specification and reported to the national building register at the NMA.

New requirements from the planning and building sector have recently emerged. This creates new demands that neither the current process nor the current national building specification can fulfil. Sweden is therefore in the process of developing a new national solution for this information. A national platform has been proposed for all information needed in the planning and building process where actors can share and use the information, and where the building information is stored in the national building register. One part of this work is to create a national specification for buildings (denoted NS building below); a draft version is currently available [38].

Figure 3 describes the proposed national solution for the exchange of building information. Data in the national building register are collected by, e.g., terrestrial surveying and aerial images/laser scanning, and in the future likely also from converted BIM models. The majority of the information is collected by municipalities, but also by the NMA (data collection layer in Figure 1), and the resulting building register is included in the building theme layer in Figure 1. This information, together with information from other themes, is provided via a national platform to consumers, who can retrieve the information as 3D city models in, for example, CityGML or CityJSON format (city model layer in Figure 1). Finally, applications use the information for their intended purpose (application layer in Figure 1).

In the case study, we evaluate how the temporal information is collected and stored in the national building register, and how that information is propagated to the national platform and then further to the applications. The process is also examined from the data providers’ point of view, mainly in terms of whether they have enough incentives to version their 3D city models and how the information is collected.

3.2. Information Gathering in the Case Study

For Issue 1, data provider incentives, a survey was sent to four cities that use 3D city models today, and three of them answered. For Issue 2, collection of versioning information, 11 municipalities were sent a survey; five municipalities decided to participate, two bigger cities and three smaller. To answer Issues 3 to 5 (database implementation on building theme layer, capability of versioning management in the building register and mapping versioning information between layers), current and proposed Swedish and international geodata standards were investigated. Finally, a review of tools and their documentation was performed to answer Issue 6, application tools.
Both surveys were sent to a limited number of municipalities, and around half of them responded. The results therefore do not give the full picture of all municipalities, but still good insight into their situation. Furthermore, all participating municipalities are members of a reference group for the new national building specification in Sweden (see Section 3.1) and are aware of its content and purpose and the proposed new model for sharing building information.

3.3. Issue 1: Data Provider Incentives

The first issue examines what incentives data providers (municipalities) have to version their 3D city models. The participating municipalities currently use their 3D city models for visualization, both to show the current city and for urban development projects. They do not store different versions of their model in a database, but one municipality saves different versions for visualization outside the database.

The municipalities foresee an increased use of 3D city models in the future, for example, for thematic visualization, visualization of alternative scenarios, noise analysis, facility management and the 3D printing of buildings. One municipality believes that it will increase the efficiency of obtaining rapid analysis and statistics of both the existing and planned urban environment, and will also make communication more efficient, as 3D city models can be made available on several platforms. For another municipality, the 3D city model will be the norm for parts of the municipality base map.

Discussion

Interviews with municipalities show that they all plan to extend their use of 3D city models and the first step for this is that their building information should conform to the new NS Building specification. This means that additional information should be provided and that major changes must be made, both in their technical solutions and in their work processes (see Section 3.4). To do this, they must have strong incentives. They could, for
example, be obliged to provide the additional information, get reimbursed for providing it, or find it useful for their own purposes.

Examples of future use of the models that the municipalities foresee are to use them for different planning scenarios, and also for some analyses. This requires that several simultaneous versions of the city are available. These versions do not have to be connected to the building register, and the municipalities can handle these versions themselves. On the other hand, 3D city models that are used for facility management or are part of the base map require links to the building register, and this information must therefore be continuously updated and reported to the national building register. Having these links will also allow for new possibilities, such as the tax authority and other authorities being able to fetch required information directly from the national platform.

3.4. Issue 2: Collection of Versioning Information

As described in Section 3.1, the Swedish building register is managed in collaboration between the municipalities and the NMA. When the implementation of the national platform for information for the planning and building sector is finalized, as described in Figure 3, the management of buildings will also change, concerning both the information content and the management routines. This section describes the current process and consequences that will arise when changing to the new process.

3.4.1. Current and Future Version Management Process of Building Information in Sweden

Table 1 describes the activities that municipalities must perform to provide lifecycle and versioning information to the NMA during the lifetime of a building. Current activities are indicated in black text in the table and the new additional activities that municipalities must provide are indicated in red. Building geometries in 3D together with additional attribute information must also be provided, but are not further described here.

Table 1. Activities that municipalities perform to provide lifecycle and versioning information to the national mapping authority (NMA) during the lifetime of a building. Additional activities and versioning information that must be provided to the new national platform are indicated in red text.

| Activity | Attributes |
|----------|------------|
| Building is planned | Status: planned |
| Building permit is accepted | Id Status: planned |
| Building is set out | Id Status: planned Geometry: standard value |
| Building is set out (Building is not registered before) | Id Status: planned Geometry: exact geometry |
| Building is set out (Building is already registered) | Geometry: exact geometry |
| Building is under construction | Status: under construction Date |
| Building is constructed and final check is performed (Building is already registered) | Status: functional Date |
| Building is constructed and final check is performed (Building is not registered before) | Id Status: functional Date Geometry: exact geometry |
| Building is extended | Status: incorrectly registered |
| Building is extended and final check is performed | Status: functional Date Geometry: modified geometry or new building part |
Table 1. Cont.

| Activity                                      | Attributes                                      |
|-----------------------------------------------|-------------------------------------------------|
| Building is reconstructed                     | Date <br> Geometry: modified geometry or new building part |
| Use of building is changed                    | Date                                            |
| Demolition of building is planned             | Status: planned demolition                      |
| Demolition permit is granted for building     | Status: demolition                              |
| Building is demolished or for another reason no longer exists | Status: deregistered Date |

Interviews were conducted in five municipalities about the current process of providing building information to the NMA, and also if and how they can provide the additional versioning information that is included in NS Building.

Evaluation of the current situation shows that either the GIS department or registration personnel are responsible for the data provision. The building information is stored in databases, and, in three municipalities, historical information can be retrieved by date from the database, but is rarely used. The others have versioning for security reasons only.

All municipalities have GIS software and storage systems for the building geometry and the attribute information currently provided to the NMA. Another software and storage system is used for case management in the building permit process, and all municipalities have different software vendors for these two systems. Results concerning the future process show that in all but one municipality, only information that currently is provided to the NMA is stored in the GIS system. The new additional temporal information to be provided is stored in the case management system and includes new dates and status codes for buildings (i.e., attributes in red text in Table 1). The case management system links this information to the property unit, not to the building. This complicates the situation, as the information must first be linked to the correct building on the property unit, and then also to the correct part of that building (Figure 4), before providing the information to the NMA.

Figure 4. Simplified description of how temporal and versioning data from the municipality must be processed to conform to national specification (NS) Building.

In all municipalities, the building permit department is responsible for the additional temporal information, as it is stored in their systems. Therefore, both new routines for
handling the information and technical solutions for providing it must be developed. Both
the GIS software and its corresponding storage must be modified to meet the new require-
ments. Municipalities see a need to retrieve historical versions of building information
from this system and foresee an increasing use if it is more easily available.

3.4.2. Discussion

All participating municipalities have the additional temporal information, but it is
difficult to retrieve it from their internal case management system and convert it to the
desired NS Building format.

As it will be both expensive and time consuming for municipalities to re-design their
technical solutions and change their work processes, they must have strong incentives to
provide the additional temporal information to the national building register. An advantage
to updating the national building register is that this additional information is required
by, e.g., the tax authorities, who then could fetch it themselves, so municipalities will no
longer have to provide it directly to them. In any case, as this entails substantial changes, it
is important that municipalities get involved in the process at an early stage.

3.5. Issue 3: Database Implementation on Building Theme Layer

Only a few 3D city models are versioned, and one reason for this could be that the
versioning method is difficult to implement, because either the method is too complex itself
or it differs from how information is currently stored in the database. How to evaluate
which changes should be implemented in the database is described in this section.

3.5.1. Implementation in a Database

The first step when implementing a versioning method in a database is to compare
its specification with the current database schema. In this case study, the versioning
information in NS Building is compared with the current database schema of the national
building register.

NS Building is influenced by INSPIRE Building and CityGML versions 2.0 and 3.0
and includes more or less the same feature types, attributes and versioning information.
Additional national information is also included. In NS Building, the features Building,
BuildingPart and BuildingInstallation inherit the feature ExchangeObject, which includes
an objectId, and versionValidFrom and versionValidTo, describing transaction time. The
feature Geometry also inherits this information, which is not the case in INSPIRE Building
or CityGML, where geometries do not include any versioning information. Additional
lifecycle information included in NS Building is attributes for legal dates: datePlanning,
dateConstruction, dateExtension, dateReconstruction, dateChangeOfUse and dateDemolition, and
the corresponding building status values (status) (see Figure 5).

The current national building register lacks the additional legal date attributes, and in
the database schema, most of the attributes are defined on the building object, while in NS
Building most of the attributes, including temporal attributes, are defined on the building
parts. Some data types also differ, and the database includes a versionId attribute.

3.5.2. Discussion

One reason why so few versioning methods are implemented could be that they are
difficult to implement in a database. Still, many lifecycle and versioning requirements from
applications must be implemented in a database (building theme layer, Figure 1) and then
propagated through the city model layer in order for it to work properly.

The complexity of the database solution differs depending on the versioning method
chosen. Providing temporal information about an object, such as dateConstruction and
dateExtension, is much simpler to implement than storing different versions of the object in
the database. This is needed, for example, to retrieve information about the building area
at a certain date, or to retrieve building geometries from different dates. It should also be
noted that once a new database schema is implemented, storing information about new
buildings should not cause any problems. What might be problematic is converting all building information that is stored according to the old database schema to the new one.

Figure 5. Part of the information model from the proposed Swedish national specification for buildings. Versioning information is marked in red boxes (model translated from Swedish by the authors).

It must also be considered whether or not temporal requirements from the applications should be implemented in the database. The requirement to synchronization BIM and geodata building models (for tracing which BIM model was converted to a certain geodata building) presupposes, for example, that both models are versioned at the database level and that the synchronization is made there. On the other hand, simultaneous alternative descriptions of buildings might not have to be saved in the database.

3.6. Issue 4: Capability of Version Management in the Building Register

The case study also evaluates whether the versioning in the proposed Swedish solution for the exchange of building information in the planning and building process (Figure 3) is sufficient for the requirements of applications in this sector. It is performed by comparing the versioning information included in NS Building with versioning requirements of
applications in the planning and building sector. Here, it is assumed that the versioning functionalities in NS Building are implemented in the national building register.

Examples of application requirements are described by Eriksson et al. [10]. Out of the requirements described in that study, three were chosen for a test case in which three versioning methods were evaluated and compared (PLCS, CityGML 3.0 and a modified GIT versioning method implemented in CityJSON). The same requirements were used in the evaluation of NS Building.

NS Building satisfies the requirements as follows:

1. Retrieve information from a specific moment in time about the following:
   - Buildings, building geometries and 3D property units in a city model (transaction time);
   - Real-world buildings and 3D property units (valid time).

   It is possible to retrieve certain legal dates from a building, that is, the dates for planning (datePlanning), construction (dateConstruction), extension (dateExtension), reconstruction (dateReconstruction), change of use (dateChangeOfUse) and demolition (dateDemolition). Information about when a building object was modified in the database (versionValidFrom and versionValidTo) can also be retrieved. Moreover, the Geometry feature inherits from the ExchangeObject feature and includes the attributes objectId, versionValidFrom and versionValidTo (Figure 4). This enables geometries to be versioned separately so that information about a geometry from a certain moment can be retrieved.

2. Have simultaneous alternative descriptions of 3D city models or buildings.

   NS Building does not include functionalities that enable simultaneous alternative descriptions of buildings, which is included, for example, in CityGML 3.0 (see Section 2.1).

3. Keep track of the synchronization between BIM and geodata models.

   The feature Building includes externalReference, which enables it to link to external sources, such as BIM models (Figure 6). To keep track of synchronization between a BIM model and NS Building, both models must have a unique and persistent identifier and be versioned. A version identifier is not included in NS Building, so this requirement cannot be met.

![Figure 6. External referencing in NS Building; the attribute is marked in a red box (model translated from Swedish by the authors).](image)

Discussion

When evaluating NS Building against the requirements from the study by Eriksson et al. [10], the results show that it can keep track of transaction time and valid time but cannot have simultaneous alternative descriptions of a 3D city model or keep track of the synchronization between BIM and geodata models. Simultaneous alternative descriptions of buildings are mainly used in analysis or to visualize alternative solutions. The software
used for analysis needs this information, but such versions do not need links to the building register and could probably be saved outside the database.

Eriksson et al. [10] concluded that there is always a trade-off between how comprehensive the version management solution is and how difficult it is to implement and use. The modified Git proposal is deliberately a simple versioning method, while PLCS is a very comprehensive lifecycle standard. It is therefore important to understand the application requirements before choosing a versioning method.

3.7. Issue 5: Mapping Versioning Information between Layers

It should be possible to map temporal and versioning information between layers, such as from the national building register to a 3D city model (see Figure 1). Whether this is possible is evaluated in this section.

3.7.1. Mapping between Layers

To evaluate whether mapping of temporal and versioning information between layers is possible, the versioning and temporal information in NS Building is compared with other current specifications for 3D city modelling. Table 2 gives an overview of similarities and differences between the versioning information included in the specifications NS Building, INSPIRE Building, CityGML 2.0, CityGML 3.0 and the proposed modified Git versioning implemented in CityJSON.

| Attribute/Specification | NS Building | INSPIRE Building | CityGML 2.0 | CityGML 3.0 | Modified Git in CityJSON |
|-------------------------|-------------|------------------|------------|------------|-------------------------|
| Version id              | versionId   |                  | gml:id     | versions   |                         |
| Transaction date from   | versionValidFrom | beginLifespanVersion | creationDate | creationDate | creationDate            |
| Transaction date to     | versionValidTo   | endLifespanVersion | terminationDate | terminationDate | terminationDate         |
| Lifespan real-world     | validFrom     |                  |            |            |                         |
| object from             |              |                  |            |            |                         |
| Lifespan real-world     |              |                  |            |            |                         |
| object to               |              |                  |            |            |                         |
| Date of planning        | datePlanning  |                  |            |            |                         |
| Date of construction    | dateConstruction | dateOfConstruction | yearOfConstruction | dateOfConstruction | dateOfConstruction |
| Date of extension       | dateExtension |                  |            |            |                         |
| Date of reconstruction  | dateReconstruction | dateOfRenovation |              |              | dateOfRenovation       |
| date of change of use   | dateChangeOfUse |                  |            |            |                         |
| Date of demolition      | dateDemolition | dateOfDemolition | yearOfDemolition | dateOfDemolition | dateOfDemolition      |
| Link to previous version|              |                  |            |            |                         |
| External reference      | externalReference | externalReference | externalReference | externalReference | parents                |

NS Building includes almost all attributes that are present in the other specifications; not included are versionId and attributes for valid dates. NS Building also includes additional legal dates (datePlanning, dateExtension and dateChangeOfUse). What NS Building does not include is versioning information similar to the versioning module in CityGML 3.0.

3.7.2. Discussion

The city model layer (Figure 1) should comply with standardized formats such as the CityGML standard and the CityJSON specification; therefore, the national building register should include the required information and be structured in a way that simplifies the mapping of information from the register to the city model. The same is also true for the
versioning capabilities. VersionId should be added to NS Building, otherwise all required information is included. An issue here is how the additional information from NS Building should be provided. This could be solved, for example, by developing a CityGML ADE or providing the information in NS Building format.

3.8. Issue 6: Application Tools for 3D City Models

The capabilities of application tools for 3D city models to handle temporal information and manage versions are also evaluated. Three tools were examined, mainly by studying freely available documentation. The evaluated application tools are ArcGIS Pro version 2.5 from Esri (https://www.esri.com/en-us/arcgis/products/arcgis-pro/resources), QGIS version 3.14 from OSGeo (https://qgis.org/en/site/about) and virtualcitySUITE version 3.1 from Virtual City Systems (https://www.virtualcitysystems.de/en/news/431-virtualcitysuite-3-1-geospatial-solution-citygml). To limit the study, the evaluation focused on temporal information and version management in tools for analysis involving city objects. The evaluation includes the following questions:

- What type of temporal information can be handled by the application tool?
- Do analyzing methods allow for 3D data or are only 2D data allowed?
- Do analyzing methods include 3D city objects?
- Is temporal information included in the analysis?
- Can time series data be used in the analysis?

3.8.1. ArcGIS Pro

In ArcGIS Pro, time properties can be set either on each feature or on a layer. Temporal information can describe a timestamp or a time interval. For continuously updated data, the option Data is a live feed can be set. Temporal data can be visualized using the Time slider, and Animation can visualize progression as a video. Bookmarks to mark a geographic location from a particular time and time properties to filter out a subset of records in a table are also available.

Tools in ArcGIS Pro are divided into toolboxes depending on the area of use. 3D Analyst is the toolbox for 3D GIS data. It provides functionalities to, for example, create and analyze 3D data and to perform visibility analysis with 3D feature and surface data. Several other analyzing capabilities are available, but they do not allow 3D or involve city objects. The Visibility toolset in 3D Analyst includes tools for the analysis of 3D city objects. Out of the 11 tools in this toolset, two use temporal information: Sun Shadow Frequency and Sun Shadow Volume. Both tools use time series data, but the temporal information concerns temporal changes in the surroundings and not in 3D city objects.

3.8.2. QGIS

In QGIS 3.13, time properties can be set as time intervals on the layer level. Temporal data can be visualized using the Temporal controller, where time progression of raster, mesh and vector data can be animated in various ways. Also, data from WMS-T services (a Web Map Service that supports temporal requests) can be visualized.

QGIS includes a Processing toolbox that includes, among other things, different types of analysis functionalities. Three-dimensional surface data such as elevation models can be used in analysis, but no analysis includes temporal information or involves 3D city objects. Analysis results that contain 3D data can be visualized in the 3D Map Viewer.

3.8.3. Virtual City Suite

Virtual City Suite is a modular web portal solution for presenting 3D city models, 3D mesh models, aerial images, point clouds and 2D maps integrated into applications. VirtualcityPLANNER is an additional module in which actors can upload data and draw, change and style the data into a 3D city model that can be shared with other users to show different urban planning scenarios, for example. XPlanGML (a German semantic data model for urban planning [12]) documents can be integrated into the 3D city model, which
makes it possible to use the model for land-use planning and preliminary building permits in Germany. Temporal information is not included in these scenarios and no analysis method is included in Virtual City Suite.

3.8.4. Discussion

At the present time, only a few application tools can handle 3D city models. Very few software tools for CityGML can read the standardized datasets correctly and fewer can export them consistently, which creates a gap between end users’ expectations and what the tools can accomplish [39]. Even fewer tools can handle temporal information in this context. The two most comprehensive pieces of GIS software have enhanced temporal capabilities, but this mainly concerns the ability to add timestamps or time intervals at the layer level and animate these changes, not managing additional temporal information at the feature level.

Application tools that currently perform analyses in urban areas use digital surface models instead of 3D city models, and the temporal information included concerns temporal changes in the surroundings (e.g., position of the sun) and not changes in city objects (e.g., an old building was demolished).

4. Recommendations for Versioning of 3D City Models

The results from the case study are summarized in Table 3, categorized by layer from the model for providing 3D city information (see Figure 1). Based on our findings, we provide recommendations on how to handle versioning of 3D city models, which are described in more detail below. It should be stressed that our approach is based on the usage of a national building register.

Table 3. Summary of issues for each layer in the model to provide 3D city information.

| Layer            | Issues                                                                 |
|------------------|------------------------------------------------------------------------|
| Data collection layer | • Data providers do not see the need to collect additional lifecycle or versioning information.  |
|                  | • Or required data:                                                   |
|                  | • Is not available;                                                  |
|                  | • Does not conform to the new structure;                              |
|                  | • Is stored in another software system and provided by another vendor; |
|                  | • Is managed by another department.                                    |
| Building theme layer | • National building register:                                         |
|                  | • Is not structured in conformance with the new standard;             |
|                  | • Does not include all information from the new standard;             |
|                  | • Does not include the version management required by the new standard.|
| City model layer | • City model layer:                                                   |
|                  | • Does not include information required by applications;              |
|                  | • Does not conform to a standardized format (e.g., CityGML standard or CityJSON specification); |
|                  | • Is not versioned.                                                   |
| Application layer | • Applications cannot retrieve required information from the city model layer. |
|                  | • Required information exists, but application requires it to be versioned. |
|                  | • Currently available software tools do not include lifecycle and versioning functionalities required by applications. |

Starting with the data collection layer, new activities must be performed and additional data provided to fulfil the requirements in the national building register (conforming to NS Building, see Table 1). All municipalities involved in the case study have the additional lifecycle information, but it is not easily available as it is stored in different systems (the building permit case management systems) that are provided by other software vendors and handled by other departments in the municipalities. Also, this information must be restructured as it is connected to property units, not to buildings. All together, this makes
it a challenge to collect and provide the required information to the national building register. To solve this, software providers should be involved from an early stage and current workflows should be modified to comply with the new conditions.

The building theme layer includes the database implementation of the national building register and should include the attributes and be structured in conformance with NS Building. Some discrepancies were found when comparing NS Building with the current database solution at the NMA. The database implementation will be more complex if there is a big discrepancy between the proposed versioning method and how information is currently stored, and if the versioning method itself is complex. Versioning at the database level makes it possible to retrieve information about a building from various phases during its lifecycle. Lifecycle information attributes (e.g., the legal dates `dateConstruction` and `dateDemolition`) can also be added to building objects. The first method is more complex to implement, but it will also fulfill more application requirements.

The city model layer should include all information required by applications in the planning and building sector together with temporal and versioning information if required. Information from this layer should be provided in a standardized format (e.g., CityGML standard or CityJSON specification). It should be possible to map temporal, versioning and other types of information between layers, for example, from the building theme layer to the city model layer. Table 2 summarizes the results from a comparison between NS Building, INSPIRE Building, CityGML 2.0 and 3.0 and the proposed modified Git versioning implemented in CityJSON. The results show that regarding temporal and versioning information, NS Building includes almost all attributes from the other specifications, and it is recommended to add `versionId`, as it is a prerequisite for synchronization between BIM and geodata models (see Section 3.5).

The application layer includes possible applications for 3D city models. In general, only a few software tools can read and export standardized datasets (e.g., CityGML) correctly [39]. The evaluation of 3D city model application tools reveals limitations in the handling of temporal information. It mainly concerns the ability to add timestamps or time intervals at the layer level, not managing temporal information at the feature level. Analyzing tools primarily use digital surface models instead of 3D city models, and do not model changes in city objects. Software developers need to be aware of these shortcomings and that the addition of lifecycle functionalities could increase the use of their software.

To conclude, we recommend linking 3D city models to the national building register. This would allow 3D city models to be used in more applications, such as building permits, 3D cadastre and facility management, and also in both municipal base maps and 3D cadastral index maps. It also enables tax authorities and other authorities to fetch the information they require directly from the city model layer (Figure 1). The chosen versioning solution should meet the established requirements, but it should be noted that more complex methods are also more difficult to implement and use.

5. Conclusions

The aim of this study is to identify obstacles to the version management of 3D city models and to provide recommendations on how to overcome these obstacles. Within this, the study has mainly focused on the municipality perspective on applications in the planning and building sector. Obstacles were identified in a case study using the implementation of a new Swedish national building standard (denoted NS Building) as an example. It is built around an architectural model with four layers: data collection, building theme, city model and application (Figure 1). The case study is largely based on the use of a national building register in the building theme layer and utilizing this register as the fundamental source for updating 3D city models.

In the case study, six issues were evaluated in connection to version management. Issue one concerns data provider incentives. The results show that both requirements from applications and strong incentives for data providers to include lifecycle and versioning information in 3D city models are essential.
The second issue relates to the collection of versioning information. When implementing the new building standard, municipalities, which are the main data providers, must collect additional lifecycle and versioning information. This requires substantial restructuring of both technical solutions and work processes. Strong incentives are therefore needed for them to invest time and money to implement such a solution to provide this information to the NMA, which is in charge of the national building register.

Issue three concerns the database implementation at the building theme layer (at the national level) and is closely related to issue four, capability of versioning management in the building register (versioning methods in the database should fulfill application requirements). That is, the NMA must restructure the storage and versioning of the national building register, but should only implement capabilities required by applications, as the complexity of the database implementation grows with the number of versioning functionalities included.

The fifth issue regards the mapping of versioning information between layers. The versioning capabilities from the building theme layer should be propagated to the city model layer, from which information in accordance with 3D city model standards/specifications (e.g., CityGML and CityJSON) is provided to various applications. The mapping of versioning information was evaluated in a comparison between NS Building and four 3D city model standards. The results are summarized in Table 2 and show that, regarding temporal and versioning information, NS Building (and with this also the national building register) includes almost all attributes from the other specifications, and it is recommended to add versionId, as it is a prerequisite for synchronization between BIM and geodata models (Section 3.5). If required, additional versioning capabilities, such as having different simultaneous versions of 3D city models, could also be added to the city model layer.

Finally, issue six relates to application tools. Applications in the application layer will use information from the city model layer for their intended purpose. To achieve this, software tools must be able to handle standardized 3D city models, including lifecycle and versioning information. Results from the evaluation of software tools show that this is not always the case and software developers need to be aware of these shortcomings and if possible add such functionalities to their software.

To conclude, we recommend linking 3D city models to the national building register, as this will simplify the versioning of 3D city models and by this also enable more complex use of the models in, for example, building permits, 3D cadastre, facility management and municipal base maps. This will also enable tax authorities and other authorities to fetch required (versioning) information directly from the city model layer (Figure 1).

**Author Contributions:** Conceptualization, Helen Eriksson and Lars Harrie; Methodology, Helen Eriksson and Lars Harrie; Formal Analysis, Helen Eriksson; Investigation, Helen Eriksson; Writing—Original Draft Preparation, Helen Eriksson; Writing—Review and Editing, Lars Harrie All authors have read and agreed to the published version of the manuscript.

**Funding:** The first author was financed by Lantmäteriet; the second author was funded by Lund University and the Smart Built Environment 3CIM project, grant number 2020-00261 (financed by Formas, Vinnova and Swedish Energy Agency).

**Acknowledgments:** We would like to thank Lantmäteriet, the Swedish mapping, cadastral and land registration authority, and Lund University for their financial support. We would also like to thank all the municipalities that responded to our surveys. Comments on a draft version by Jesper Paasch are greatly appreciated.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Biljecki, F.; Stoter, J.; LeDoux, H.; Zlatanova, S.; Çöltekin, A. Applications of 3D City Models: State of the Art Review. *ISPRS Int. J. Geo-Inf.* 2015, 4, 2842–2889.
2. Julin, A.; Jaalama, K.; Virtanen, J.-P.; Pouke, M.; Ylipulli, J.; Vaaja, M.; Hyyppa, J.; Hyyppä, H. Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS Int. J. Geo-Inf.* 2018, 7, 55.
3. Gröger, G.; Kolbe, T.H.; Nagel, C.; Häfele, K.H. (Eds.) OGC 12-019 OpenGIS City Geography Markup Language (CityGML) Encoding Standard. 2012. Available online: http://www.opengeospatial.org/standards/citygml (accessed on 23 September 2020).
4. Chaturvedi, K.; Smyth, C.S.; Gesquière, G.; Kutzner, T.; Kolbe, T.H. Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML. In Advancing Geoinformation Science for a Changing World; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2016; pp. 191–206.
5. Kutzner, T.; Chaturvedi, K.; Kolbe, T.H. CityGML 3.0: New Functions Open Up New Applications. Photogramm. Fernerkund. Geoinf. 2020. [CrossRef]
6. INSPIRE. Data Specification on Buildings—Technical Guidelines, Version 3.0, European Commission Joint Research Centre. 2013. Available online: http://inspire.ec.europa.eu/id/document/tg/bu (accessed on 23 September 2020).
7. Vitalis, S.; Labetski, A.; Arroyo Ohori, K.; Ledoux, H.; Stoter, J. A data structure to incorporate versioning in 3D city models. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2019, 4, 123–130.
8. ISO 10303-239:2012: Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 239: CityGML data model. Available online: http://www.opengeospatial.org/standards/citygml (accessed on 22 December 2020).
9. ISO 19152:2012: Geographic Information—Land Administration Domain Model (LADM). 2012. Available online: https://www.iso.org/standard/51206.html (accessed on 22 December 2020).
10. Eriksson, H.; Sun, J.; Tarandi, V.; Harrie, L. Comparison of versioning methods to improve the information flow in the planning and building processes. Trans. GIS 2020. [CrossRef]
11. Olsson, P.-O.; Axellsson, J.; Hooper, M.; Harrie, L. Automation of Building Permission by Integration of BIM and Geospatial Data. ISPRS Int. J. Geo-Inf. 2018, 7, 307.
12. Benner, J.; Geiger, A.; Häfele, K.H. Concept for Building Licensing based on standardized 3D Geo Information. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Berlin, Germany, 3–4 November 2010; Volume XXXVIII-4/W15, pp. 9–12.
13. Van Berlo, L.; Dijkmans, T.; Stoter, J. Experiment for integrating Dutch 3D spatial planning and BIM for checking building permits. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2013, 2, 279–284.
14. Eirinakia, M.; Dharb, S.; Mathurc, S.; Kaleya, A.; Patelc, A.; Joshua, A.; Shahc, D. A building permit system for smart cities: A cloud-based framework. Comput. Environ. Urban Syst. 2018, 70, 175–188.
15. Noardo, F.; Ellul, C.; Harrie, L.; Overland, I.; Shariat, M.; Ohori, K.A.; Stoter, J. Opportunities and challenges for GeoBIM in Europe: Developing a building permits use-case to raise awareness and examine technical interoperability challenges. J. Spat. Sci. 2020, 65, 209–233.
16. El-Mekawy, M.S.A.; Paasch, J.M.; Paulsson, J. Integration of possible legal aspects in 3D cadastral systems. Int. J. E-Plan. Res. 2015, 4, 47–71.
17. Gröger, G.; Kolbe, T.H.; Nagel, C.; Häfele, K.H. (Eds.) OGC 12-019 OpenGIS City Geography Markup Language (CityGML) Encoding Standard. 2012. Available online: http://www.opengeospatial.org/standards/citygml (accessed on 23 September 2020).
18. Chaturvedi, K.; Smyth, C.S.; Gesquière, G.; Kutzner, T.; Kolbe, T.H. Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML. In Advancing Geoinformation Science for a Changing World; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2016; pp. 191–206.
19. Kutzner, T.; Chaturvedi, K.; Kolbe, T.H. CityGML 3.0: New Functions Open Up New Applications. Photogramm. Fernerkund. Geoinf. 2020. [CrossRef]
20. INSPIRE. Data Specification on Buildings—Technical Guidelines, Version 3.0, European Commission Joint Research Centre. 2013. Available online: http://inspire.ec.europa.eu/id/document/tg/bu (accessed on 23 September 2020).
21. Vitalis, S.; Labetski, A.; Arroyo Ohori, K.; Ledoux, H.; Stoter, J. A data structure to incorporate versioning in 3D city models. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2019, 4, 123–130.
22. ISO 10303-239:2012: Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 239: CityGML data model. Available online: http://www.opengeospatial.org/standards/citygml (accessed on 22 December 2020).
23. Worboys, M.F.; Duckham, M. GIS: A Computing Perspective, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2004; Chapter 10.
24. Worboys, M.F. Event-oriented approaches to geographic phenomena. Int. J. Geogr. Inf. Sci. 2005, 19, 1–28.
25. Wieland, M.; Pittore, M. A spatio-temporal building exposure database and information life-cycle management solution. ISPRS Int. J. Geo-Inf. 2017, 6, 114.
26. Liu, X.; Wang, X.; Wright, G.; Cheng, J.C.P.; Li, X.; Liu, R. A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographical Information System (GIS). ISPRS Int. J. Geo-Inf. 2017, 6, 53.
27. OGC. CityGML 3.0 Conceptual Model. 2019. Available online: https://github.com/opengeospatial/CityGML-3.0CM (accessed on 23 September 2020).
28. Directive 2007/2/EC of the European Parliament and of the Council Establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). 2007. Available online: http://inspire.ec.europa.eu/documents/directive-20072ec-european-parliament-and-council-14-march-2007-establishing (accessed on 23 September 2020).
29. Ledoux, H.; Arroyo Ohori, K.; Kumar, K.; Dukai, B.; Labetski, A.; Vitalis, S. CityJSON: A compact and easy-to-use encoding of the CityGML data model. Open Geospat. Datasoftw. Stand. 2019, 4, 4.
30. Nguyen, S.H.; Yao, Z.; Kolbe, T.H. Spatio-semantic comparison of large 3D city models in CityGML Using a Graph Database. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2017, IV-4/W5, 99–106.

31. Nguyen, S.H.; Kolbe, T.H. A multi-perspective approach to interpreting spatio-semantic changes of large 3D city models in CityGML using a graph database. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2020, VI-1/W1-2020, 143–150.

32. Olsen, B. Integration of 3D City Models in a Country Wide Covering 3D Basemap. Master’s Thesis, Delft University of Technology, Delft, The Netherlands, 2019.

33. Hijazi, I.H.; Krauth, T.; Donaubauer, A.; Kolbe, T. 3DCITYDB4BIM: A system architecture for linking BIM server and 3D citydb for BIM-GIS-integration. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2020, V-4-2020, 195–202. [CrossRef]

34. Yao, Z.; Nagel, C.; Kunde, F.; Hudra, G.; Willkomm, P.; Donaubauer, A.; Adolphi, T.; Kolbe, T.H. 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospat. Data Softw. Stand.* 2018, 3, 5. [CrossRef]

35. Agugiaro, G. First steps towards an integrated CityGML-based 3D model of Vienna. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2016, III-4, 139–146. [CrossRef]

36. Skarbal, B.; Peters-Anders, J.; Faizan Malik, A.; Agugiaro, G. How to pinpoint energy-inefficient buildings? An approach based on the 3D city model of Vienna. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2017, IV-4/W3, 71–78.

37. Doulamis, A.; Soile, S.; Doulamis, N.; Chrisouli, C.; Grammalidis, N.; Dimitropoulos, K.; Manesis, C.; Potsiou, C.; Ioannidis, C. Selective 4D modelling framework for spatialtemporal land information management system. In Proceedings of the SPIE 9535, Third International Conference on Remote Sensing and Geoinformation of the Environment, Paphos, Cyprus, 19 June 2015. [CrossRef]

38. Nationell Informationsspecifikation Byggnad, Version 1. 2019. Available online: https://www.lantmateriet.se/contentassets/ade15b08cbfe43fc95b83e3f28c281d/natspec-byggnad-v1.0.pdf (accessed on 23 September 2020).

39. Noardo, F.; Arroyo Ohori, K.; Biljecki, F.; Ellul, L.; Harrie, L.; Krijnen, T.; Eriksson, J.; van Liempt, J.; Pla, M.; Ruiz, A.; et al. Reference Study of CityGML Software Support: The GeoBIM Benchmark 2019—Part II. *Trans. GIS* 2020. (under review). Available online: https://arxiv.org/pdf/2007.10938.pdf (accessed on 23 September 2020). [CrossRef]