A Data Structure for Scientific Models of Historical Cities: Extending the CityJSON Format

Beatrice Vaienti
Digital Humanities Laboratory
EPFL, Lausanne, Switzerland

Paul Guhennec
Digital Humanities Laboratory
EPFL, Lausanne, Switzerland

Isabella di Lenardo
Institute for Area and Global Studies
EPFL, Lausanne, Switzerland

ABSTRACT

In the field of the 3D reconstruction of cities in the past there is a raising interest in the creation of models that are not just geometrical, but also informative, semantic and georeferenced. Despite the advancements that were done in the historical reconstruction of architecture and archaeology, the solutions designed for larger scale models are still very limited. On the other hand, research on the digitisation of current-day cities provides useful instruments. In particular, *CityJSON* - a JSON encoding of CityGML - represents an easy-to-use and lightweight solution for storing 3D models of cities that are geolocated, semantic and that contain additional information in the form of attributes. This contribution proposes (1) to extend the schema to the needs of a historical representation; and (2) to incorporate the newly created model in a continuous flow pipeline, in which the geometry is dynamically updated each time an attribute is changed, as a means to foster collaboration.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**; Users and interactive retrieval; **Document representation**; • **Applied computing** → *Architecture (buildings)*; *Digital libraries and archives.*

KEYWORDS

historical modelling, architectural reconstruction, 4D cities, digitisation, version controlling, uncertainty visualisation, historical validation

ACM Reference Format:

Beatrice Vaienti, Paul Guhennec, and Isabella di Lenardo. 2022. A Data Structure for Scientific Models of Historical Cities: Extending the CityJSON Format. In *The 6th ACM SIGSPATIAL International Workshop on Geospatial Humanities (GeoHumanities '22) (GeoHumanities '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557919.3565813

1 INTRODUCTION

1.1 4D models as scientific tools

4D models of cities represent not only a powerful tool for dissemination and visualisation, but also have the potential to become scientific instruments. An ***informative*** model (i.e. accompanied by the explanation of the interpretative processes hidden behind and used sources) should also be seen as a document to be exchanged and commented between researchers. In that way, the reconstruction of architecture can become an analytical means of testing different hypotheses and not only an instrument to produce a final synthesis of the results [4]. Similarly, a model shared between researchers should be ***dynamic***, in other words, capable of easily adapting itself to continuously updated new pieces of information [6].

The current discourse in Digital Heritage has attracted attention on the scientific rigour required in the modelling process of Cultural Heritage. Most importantly, its transparency is in fact fundamental to validate the reconstruction hypotheses, since it enables researchers to review the result while being aware of the assumptions that led to its development [9], [3].

Although only at the scale of single buildings (and often in experimental archaeology), there are some examples that go in this direction [5], [2]. Within research on historical reconstruction of cities at the larger scale however, the discourse is not as developed, despite the great interest shown for the creation of such models.

1.2 Representation of present-day cities

Focusing on the current efforts on representing cities as they are today, we notice how similar the research questions are to those asked by historical representations. In fact, the research about the creation of 3D databases of cities tackles the increasing need expressed by urban planners for the sustainable digitisation of geospatial features. This need stems from the multiple advantages that a semantic 3D model of a city (i.e. a virtual model of the urban environment containing, besides the geometry, additional attributes like the object type) can bring to planners for optimisation and support to decision-making processes.

Research on the creation of "Urban Digital Twins" led to the CityGML standard, issued by the Open Geospatial Consortium (OGC), and upon that, 3DCityDB was developed to facilitate the deployment of such models through the creation of 3D geo-databases[1].

1.3 The CityJSON format

In particular, the recent development of a JSON encoding of CityGML - CityJSON [8] - represents a promising lightweight and easy solution for creating ***semantic*** and ***geo-referenced*** 3D city models, while still maintaining the interoperability with CityGML and 3DCityDB. Interestingly, CityJSON was also designed having in mind the necessity of editing part of the city model through versioning: this can be seen as a problem parallel to the one we encounter when new sources require modifications of an historical model.

[1]https://www.3dcitydb.org/3dcitydb/

GeoHumanities '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9533-5/22/11.
https://doi.org/10.1145/3557919.3565813

Another significant advantage is the compatibility of CityJSON with several open-source softwares, such as Blender and QGIS (through the plugins Up3Date[2] and CityJSON Loader[3] respectively). Its suitability to web applications is shown with the software NINJA, a 3D geometrical web-viewer that permits to investigate the model's semantics and attributes[4].

Finally, let us mention that CityJSON was built with the addition of extensions to the core data model in mind: the conversion of extensions' structures into the new schema is done automatically.

1.4 Contributing with a framework for the historical evolution of cities

This prompted us to think that an extension to CityJSON was an optimal solution for the creation of historical models of cities. We've identified two main gaps in the current possibilities of CityJSON:

(1) Being able to model the historical evolution of one city. In other words, being able to specify the dates of existence and shape of past and present buildings.
(2) The uncertainty associated with geometries. Unlike cities of today, the representation of a building in a historical model may present issues of uncertainty and conflicting information (both at geometrical and temporal levels). Leveraging CityJSON's multiplicity of Levels of Details (LODs) provides an interesting approach, not just for 3D viewing optimisation, but also for visualising the heterogeneity of information and reconstruction precision. Equally important is the flexibility of the model, which must easily deal with the inclusion of new sources and the modification of attributes.

This paper presents the objectives and first steps of that historical extension to CityJSON, and will follow a two-fold structure. Firstly, in Section 2, we will present the initial set of attributes deemed necessary for a simple study-case: the reconstruction of the town of Sion (Switzerland) in 1640, 1740, and 1840 using historical cartography, and the adding to the geometries some pre-digitised information about the owners. In Section 3, we will detail the idea of a continuous flow pipeline: that is, the capability of the 3D model to digest added information for the modelling in itself, without human intervention. Finally, we suggest future improvements and avenues of work in Section 4.

2 STRUCTURE OF A CITYJSON HISTORICAL EXTENSION

2.1 Object attributes

The core idea of the proposed *historicalCityJSON* structure is that the 4D representation of a building (and by extension, the city) can be modelled using only the **combination of a 2D footprint and attributes**. At the scale of the city, (historical) cartography is the natural starting point for modelling projects, with footprints from the maps being the units onto which extra information is attached. Though envisioned for the automatic modelling of simple architecture, our model remains flexible by being able to store manual models for more complicated edifices, as will be described further down. Each object is stored under a specific unique ID, which we can arbitrarily choose. The attributes fall into two categories: (1) geometrical attributes, which are all parametric values needed by the procedural modelling part of the pipeline to create a 3D object from the footprint (ie. LOD≥1 from LOD0); and (2) non-geometrical attributes, which enrich the 3D object with the contextual information required by an historical knowledge system, specifying foremost the building time boundaries.

2.1.1 *Geometrical attributes.*

In first approach, we reduced the geometrical attributes to those needed by LOD2: the height and roof components. While the former is a single numeric value (stored under ***geomAttributes.height***), the latter is subdivided into the type of the roof and its generation parameters (respectively ***geomAttributes.roof.type*** and ***geomAttributes.roof.parameters***). For now, roof types are limited to flat and hipped (in which case ***parameters*** simply contains the slope value); but extending to more types is planned[5]. It is then up to the procedural modelling software (Grasshopper) to combine the footprint with the height, the roof type and its parameters to produce a coherent 3D object.

2.1.2 *Non-geometrical attributes.*

While take its footprint as the core identity of a building, we might want to acknowledge that despite differences in footprints, two objects are conceptually connected. This is done by storing, under the non-geometrical ***nonGeomAttributes.linkedEntities*** attribute, the list of IDs of those linked objects and specify for each the nature of the link, which we divide into three categories:

(1) **Synchronic link**. Children buildings that are physically independent yet relate to the same parent conceptual entity: for example a garden and the house, different subparts of a fortification. This relates to the well-established problem of partonomy [7].
(2) **Diachronic link**. Translation or rebuilding, whereby a same 'conceptual' building has its geometrical extent modified.
(3) **Methodological link**. Conflicting hypotheses with regards to the shape of a building.

Secondly, we define ***nonGeomAttributes.parcels*** to store cadastral information as an array of the parcels pertaining the building. Under each parcel it is then possible to specify a parcel ID, the owners and the respective time spans (start and end).

Finally, and perhaps most importantly for a data format which aims at encoding the time evolution of a city, each object has its temporal existence specified. This avoids the need for as many files as there are states of the city, provided that the visualiser permits to filter the buildings to display according to the value of that field. The suggested ***time*** attribute has the following sub-attributes:

- ***nonGeomAttributes.time.estimatedStart***. This field refers to an hypothetical date of first appearance, as can be for example suggested for ancient buildings based on stylistic features.
- ***nonGeomAttributes.time.appearances***. In that field are entered all documented evidences pertaining a given building. The visualiser shall extract from this list the earliest

[2]https://github.com/cityjson/Up3date
[3]https://github.com/cityjson/cityjson-qgis-plugin
[4]https://github.com/cityjson/ninja

[5]For upcoming models, the ***parameters*** attribute will contain information other than the slope.

and latest dates to understand what is the temporal range of every object.

- ***nonGeomAttributes.time.estimatedEnd***. Here should be entered the end date of the building presence, when it is hypothetical.

The fields ***estimatedStart*** and ***estimatedEnd*** convey a sense of temporal uncertainty, and will be used by the web viewer to display the building accordingly (for example, by making it slightly transparent).

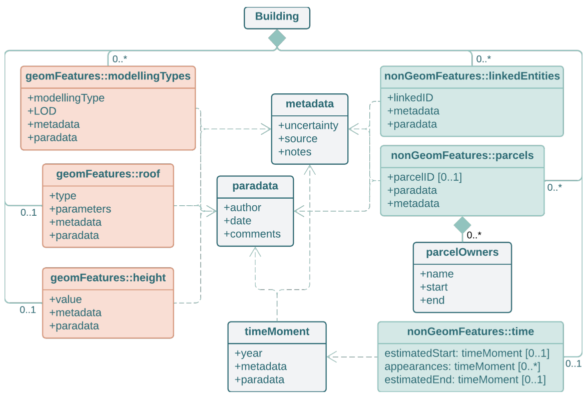

Figure 1: Proposed *historicalCityJSON* ULM schema.

2.1.3 Metadata and paradata. On top of these attributes, an extra layer of meta-information is added, a needed addition if we want this historical information system to be methodologically sound: each piece of information used in the visualiser should either be sourced or marked as unsourced. Additionally, since this structure is planned with collaboration between researchers in mind, we include fields to specify who is the author of the suggestion or edit (***paradata.author***), when that change was made (***paradata.date***), and additional spaces for comments, discussion, or approval (***paradata.comments***). The division of this extra layer between metadata and paradata echoes widely adopted practices and terminology in Digital Heritage [3]. While paradata attributes are self explanatory, the metadata attributes are as follows:

- ***metadata.uncertainty***. This numerical value, which derives from pre-established criteria (which will be described in Section 3.2), is used by the viewer to highlight the extent to which the different parts of the reconstruction are hypothetical. We propose to have such an uncertainty value for every geometrical attribute, so that the granularity of the uncertainty can depend on the LOD. Non-geometrical attributes are not given an uncertainty.
- ***metadata.source***. All sources are listed under that field[6], following an homogeneous format of reference.
- ***metadata.notes***. This text field is not used for the modelling in itself (at least yet, within the possibilities of current procedural modules), but is needed to keep track of the reasoning that can emerge from the interpretation of sources.

[6]There is one exception: since the footprint is not derived from attributes, its source is stored separately, under the LOD0 field in the Geometry tree of the file.

3 DYNAMIC REGENERATION OF 4D MODELS

3.1 Procedural modelling

In the *historicalCityJSON* extension we propose, the LOD1 and LOD2 geometries are automatically created from the LOD0 and attributes, by inputting both in a Grasshopper script which extrudes the footprint by the given ***height*** before solving the shape of the roof, according to its type and parameters. The results of this on a simple case study are shown in Figure 2.

While the main idea of the framework is to handle procedural generation, we might also want to include manual models of buildings of special architectural importance (which can come from existing projects or databases). In such cases, we set the ***modellingType*** attribute to "manual", such that only the buildings labelled as procedural are regenerated. The other attributes used for reconstruction (***nonGeomAttributes.time***, and ***nonGeomAttributes.sources***) are still valid. Note that this tag is present for every LOD geometry; indeed, we might have a LOD3 manual model while still wanting LOD2 to be procedurally generated.

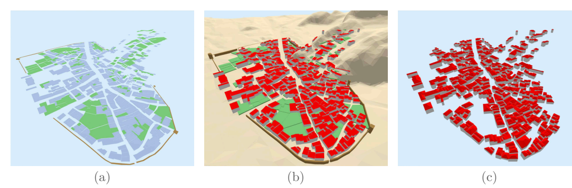

Figure 2: Procedural model of the Swiss city of Sion in 1840. (a) LOD0; (b) LOD1 and terrain elevation; (c) LOD2. The colour corresponds to the type of geometrical attribute (semantics).

3.2 Defining uncertainty

Furthermore, we aim for a scientific database visualiser, i.e. a mode of displaying the degree of certainty of each showed element. We are indeed aware of the pitfall that, without taking some precautions, 3D models can be easily perceived as truths and not as results of an (open-to-critique) interpretative process.

3.2.1 Geometrical uncertainty: What? The following classification (adapted from well-established matrices in architectural reconstruction [1]) is proposed, in increasing order of uncertainty:

- **1: Visual sources**. In cases of conflicting representations, it is up to the annotator to decide which (if any) source ought to be discarded; although we expect this scenario to be unlikely for the level of detail of city-scale reconstructions.
- **2: Secondary sources**.
- **3: Educated guess** (coeval buildings, stylistic similarity).
- **4: Random value**.

As stated in Section 2.1.3, an uncertainty value is attached to each geometrical attribute. The uncertainty of the procedurally generated geometrical elements is derived from those of used attributes: either trivially when only one attribute is used (for example, the LOD1 extrusion only uses the ***height*** attribute); or by taking the maximum uncertainty when several attributes are combined (as is the case for the roof, which uses both ***roof.parameters*** and ***roof.type***).

3.2.2 Existence uncertainty: When? Time uncertainty has been tackled in the time attribute by separating the first (or last) documented occurrence from the actual construction (or destruction). It is that distinction that we make use of to display temporal uncertainty.

4 FUTURE WORK

Although the objectives and general structure of the *historicalCityJSON* have been established, there remains a lot of work to turn it into the general, scientific, and open-access solution for diachronic reconstructions of cities we aim at.

(a) Transparent implementation. The first avenue of improvement would be to get rid of the need for an external procedural modelling tool. Our first proofs-of-concept still rely on inputting the CityJSON file into Grasshopper to generate 3D objects. Having everything self-contained in a unique Python script will require the re-implementation of geometrical modelling itself; such that one can generate the expected volume for any footprint and attribute combination. The Blender software has the advantage of being open-source and fully scriptable in Python. Some pre-existing procedural modules in Grasshopper need to be re-implemented from scratch in Blender but work in that direction has started with the first simple roof modules, and we expect to continue incrementally as needs and desires extend.

(b) Additional attributes. This goes hand-in-hand with the addition of more attributes, both geometrical and non-geometrical. For example, the development of the attributes ***geomAttributes.floor*** and ***nonGeomAttributes.usageType*** is underway. This format was started with a project that required some limited attributes, but we foresee the need for new attributes which other research projects will reveal. We hope that this permanent extending of the possibilities will foremost come from the community and, in that way, foster collaboration.

(c) Leveraging linked entities. Combining the ***uncertainty*** attribute with ***linkedEntities*** offers a powerful analysis tool on the state of knowledge on one building. What the ***linkedEntities*** attribute effectively yields is as many disconnected graphs as there are "conceptual" (separated) entities. For each of these sub-graphs, it will become possible to infer cross-contained information on amounts of interpretative conflicts or to calculate set operations from different sources and on different attributes (say, for example, the intersection probabilities of two footprints; or the building version which has the lowest geometrical uncertainty).

(d) Data structure optimisation. A strong advantage of CityJSON is its lightness (a precious quality for Web viewing) which comes from the fact that all unique vertices are given a callable ID used in the geometries, avoiding the duplication of points. Doing the same for any field that is susceptible to be repeated (including ***metadata.source***, or ***paradata.author***) will not only lighten the file but also homogenise those fields. Such homogeneity in the source field could allow a better control and feedback on the information pipeline: for example the possibility of disregarding one source and its derived attributes, if is deemed too imprecise *a posteriori*.

(e) Git versioning. Collaborative projects, as intended here, greatly benefit from a versioning solution which allows to monitor contributions, identify conflicts, and establish branches when no agreement can be found. Implementing a Git support for *historicalCityJSON* will be a simple solution to enhance the collaborativeness of it (thanks to the Pull, Push, Commit, Fork, Branch features).

(f) End-to-end pipeline. We're aware that filling every field to the extent suggested here would be a tedious operation. And indeed, *historicalCityJSON* was conceived with extended automation in mind; that is, the idea that it some information fields can be extracted using automatic pipelines (notably historical map processing and OCR techniques) if sources are following a rigid-enough format.

Beyond automatically extracted features, warping everything in an interface would facilitate the filling of some field through, for example the modification of the attributes of several buildings at once; the suggestion of values from a list; or even an author login.

Finally, the extensibility of CityJSON web viewers (such as Ninja), through relatively easy JavaScript tweaks is another advantage to leverage in the end part of the pipeline, in which we devise ways of visualising both geometrical and non-geometrical information contained in our model; and one that complements well the incremental approach we suggest for the development of new attributes. Planned extensions will, in particular, allow:

- the colour-coding of buildings according to the computed ***uncertainty*** ;
- the highlighting some geometries according to some query;
- showing the nature of connections between linked entities;
- a dynamic evolution of the city, by using the ***time*** attribute.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945363; and from the Swiss National Science Foundation [FNS/SNF DIV 2 Parcels of Venice, 2019-2023, Project No 514448].

REFERENCES

[1] Fabrizio Apollonio. 2016. *Classification Schemes for Visualization of Uncertainty in Digital Hypothetical Reconstruction.* 173–197. https://doi.org/10.1007/978-3-319-47647-6_9
[2] Irene Bassoli, Federico Fallavollita, and Wladek Fuchs. 2022. Hypothesis of reconstruction of the Roman theater of Urbs Salvia. *SCIRES-IT - SCIentific RESearch and Information Technology* 12, 1 (June 2022). https://doi.org/10.2423/i22394303v12n1p151
[3] Víctor Manuel López-Menchero Bendicho. 2013. *International Guidelines for Virtual Archaeology: The Seville Principles.* Springer International Publishing, Cham, 269–283. https://doi.org/10.1007/978-3-319-01784-6_16
[4] Emanuel Demetrescu. 2018. *Virtual Reconstruction as a Scientific Tool: The Extended Matrix and Source-Based Modelling Approach.* 103,104.
[5] Daniele Ferdani, Emanuel Demetrescu, Marco Cavalieri, Gloriana Pace, and Sara Lenzi. 2020. 3D Modelling and Visualization in Field Archaeology. From Survey To Interpretation Of The Past Using Digital Technologies. *Groma. Documenting archaeology* 15 (April 2020). https://doi.org/10.12977/groma26
[6] Frédéric Kaplan and Isabella di Lenardo. 2017. Big Data of the Past. *Frontiers in Digital Humanities* 4 (May 2017). https://doi.org/10.3389/fdigh.2017.00012
[7] C. Maria Keet and Alessandro Artale. 2008. Representing and Reasoning over a Taxonomy of Part-Whole Relations. *Appl. Ontol.* 3, 1–2 (jan 2008), 91–110.
[8] Hugo Ledoux, Ken Ohori, Kavisha Kumar, Balázs Dukai, Anna Labetski, and Stelios Vitalis. 2019. CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data Software and Standards* 4 (06 2019), 4. https://doi.org/10.1186/s40965-019-0064-0
[9] UNESCO. 2003. UNESCO Charter on the Preservation of the Digital Heritage. In *Intergovernmental Council for the Information for All Programme, 2nd, Paris.*