# TESTING THE NEW 3D BAG DATASET FOR ENERGY DEMAND ESTIMATION OF RESIDENTIAL BUILDINGS

Camilo León-Sánchez*, Denis Giannelli, Giorgio Agugiaro, Jantien Stoter

3D Geoinformation group, Delft University of Technology, Faculty of the Built Environment and Architecture,
Department of Urbanism, Julianalaan 134, Delft 2628BL – The Netherlands
{c.a.leonsanchez, d.giannelli, g.agugiaro, j.e.stoter}@tudelft.nl

**Commission IV, WG IV/10**

**KEY WORDS:** 3D BAG, CityGML, SimStadt, CitySim, Energy ADE, 3D City Database.

**ABSTRACT:**
The 3D BAG v. 2.0 dataset has been recently released: it is a country-wide dataset containing all buildings in the Netherlands, modelled in multiple LoDs (LoD1.2, LoD1.3 and LoD2.2). In particular, the LoD2.2 allows differentiating between different thematic surfaces composing the building envelope. This paper describes the first steps to test and use the 3D BAG 2.0 to perform energy simulations and characterise the energy performance of the building stock. Two well-known energy simulation software packages have been tested: SimStadt and CitySim Pro. Particular care has been paid to generate a suitable, valid CityGML test dataset, located in the municipality of Rijssen-Holten in the central-eastern part of the Netherlands, that has been then used to test the energy simulation tools. Results from the simulation tools have been then stored into the 3D City Database, additionally extended to deal with the CityGML Energy ADE. The whole workflow has been checked in order to guarantee a lossless dataflow.
The paper reports on the proposed workflow, the issues encountered, some solutions implemented, and what the next steps will be.

## 1. INTRODUCTION

Semantic 3D city models are being generated and adopted more and more by municipalities as they can represent – spatially and thematically – all most relevant urban features, e.g. ranging from buildings to vegetation, transportation, relief, water bodies, etc.
In March 2021, the 3D Geoinformation Group at TU Delft published a new country-wide dataset, called 3D BAG v. 2.0, containing all buildings in the Netherlands in three different levels of detail (LoD1.2, LoD1.3 and LoD2.2) (Stoter *et al.*, 2020). The new dataset contains circa 10 million buildings and is a major improvement over the previous version which offered only LoD1 buildings (Dukai *et al.*, 2019). All buildings are generated by combining two country-wide open datasets: the BAG and AHN. The BAG (Basisregistratie Adressen en Gebouwen) contains, among the rest, all building footprints, the addresses and some other information. The height information comes from the AHN (Actueel Hoogtebestand Nederland), a Lidar-based point-cloud dataset which is acquired on average every 4 years over the whole country. The 3D BAG dataset is available as open data and can be downloaded in different formats (e.g. CityJSON, GeoPackage and OBJ) from the web-based 3D Viewer (Figure 1, bottom). As such, it also represents a new and ideal test dataset to further test and experiment with the data, and to develop urban applications based on it (Doan *et al.*, 2021).

In this paper we describe the first tests and findings of using the 3D BAG dataset to perform energy-related analyses, such as the estimation of solar irradiance and the energy demand for buildings. The focus here is *not* on developing such applications, but to first test the quality and suitability of the 3D BAG data for this purpose. For this reason, in this first stage, existing software packages will be used, focussing on the respective data requirements and the generated outputs. Eventually, newly generated energy-related data could be

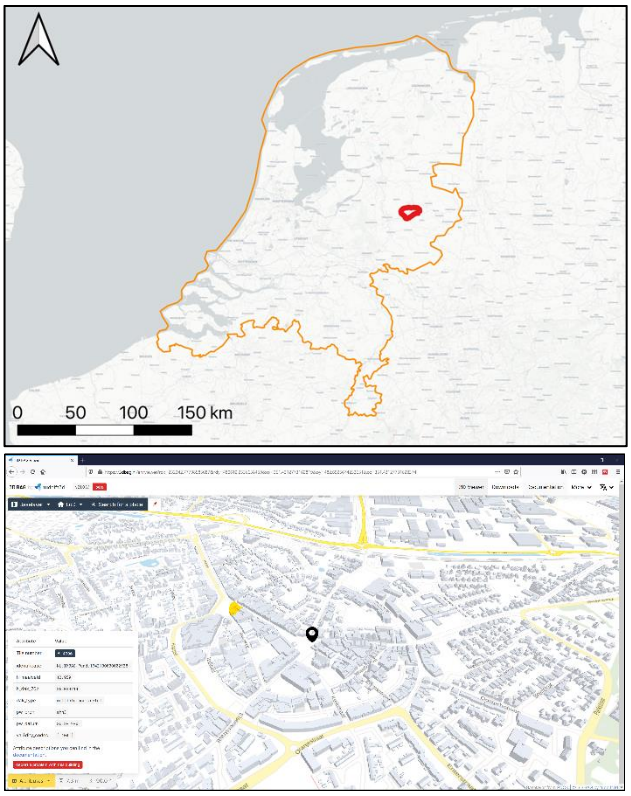


**Figure 1.** [Top] Position of Rijssen-Holten in the Netherlands. The orange line indicates the continental boundary of the country, the red line indicates the approximate position of the municipality. [Bottom]Screenshot taken from the 3D BAG viewer showing a portion of the municipality of Rijssen-Holten in the Netherlands.

* Corresponding author

https://doi.org/10.5194/isprs-archives-XLVI-4-W1-2021-69-2021

integrated “back” into the BAG 3D dataset, in order to provide a richer dataset for further applications.

The methodology presented in this paper is divided into three main steps that will be discussed sequentially in the next sections. First, datasets containing spatial and non-spatial data are prepared, checked and integrated in order to generate a valid CityGML-complaint model of the test area. Secondly, the datasets are checked for compliance with two energy simulation software packages and adapted if necessary, depending on the respective input data requirements. Finally, data output from the two simulation software packages are again tested for validity before being imported into the 3D City Database. Given the energy-based nature of the output results, the compatibility of the 3D City Database itself for storage of CityGML Energy ADE (Agugiaro *et al.*, 2018) contents is also tested.

## 2. DATA PREPARATION

The test site corresponds to the Municipality of Rijssen-Holten, a small town of circa 37000 inhabitants (and circa 22000 buildings) located in the eastern-central part of the Netherlands (Figure 1, top). CityJSON-based data of the test site was retrieved from the 3D BAG webpage (Figure 1, bottom) and converted into XML-based CityGML in order to be used in the following energy software packages. In the process, the dataset was further enriched with data coming from other available data sources like the ones shared by the municipality of Rijssen-Holten. Such datasets contain thematic information regarding for example the number of storeys, the building usage and wall cavity – the latter specifies the type of façade insulation of buildings for circa 11000 building ids.

Another source of information is the “standard” BAG dataset that, unlike the 3D BAG dataset, also includes the “gebruiksdoel” attribute, i.e. the usage of the building. However, this is a rather general classification: for example, many buildings are classified as “overige gebruiksfunctie” (in English: Other usage function) which includes garage boxes, parking garages, pumping stations, water purification buildings, water towers, gas distribution stations – just to mention some (Praktijkhandleiding BAG, 2021). It is here relevant to mention that a correct classification of the building stock according to its function/usage plays a major role in later steps of the workflow, as several other parameters needed for energy simulation are then retrieved from libraries depending on the year of construction and the building function (e.g. TABULA available at https://episcope.eu).

All operations carried out on the input datasets can be grouped into two major groups: geometry processing, and dataset enrichment by means of thematic attributes. All operations described in the following paragraphs were carried out by means of some workbenches in FME (Safe Software, 2021).

### 2.1 Geometry processing

In terms of geometrical modelling of the building stock, the sole source of information was the 3D BAG v. 2.0. CityJSON files containing buildings in LoD2.2 and covering the whole municipality area were downloaded and merged into a single CityGML dataset. As mentioned in the documentation website of the 3D BAG (3D BAG, 2021), buildings are all modelled as single-part buildings. This applies also to multi-part buildings, they are then represented as a plain list of buildings all having the same gml:id (which, as a side-effect, leads to an invalid document). For this reason, multi-part buildings were identified, reclassified as CityGML *BuildingParts*, and grouped into *Buildings*. Unique gml:ids were assigned in this case only to the root Building object containing the parts.

All geometries in the CityJSON files are triangulated surfaces. Although this is not an error per se, for each thematic surface a mesh simplification was first carried out in order to obtain, as far as possible, planar equivalent polygons instead of sets of coplanar triangles, thus reducing the number of geometry primitives. In addition, for each surface the values for sloped area, orientation (azimuth) and inclination were computed and stored as generic attributes. Further tests were carried out regarding the thematic surfaces composing the LoD2 model (i.e. GroundSurfaces, WallSurfaces, RoofSurfaces, etc.). Some errors were found (and corrected), such as:

- Missing GroundSurfaces in a (very) limited number of Building(parts). The missing geometries were generated and added by projecting and dissolving the sloped roof surfaces of the affected building;
- Thematic surfaces missing a classification or being classified incorrectly (e.g. as InteriorWallSurfaces instead of WallSurfaces). (Re)classification rules were defined based on the normal vector of each surface.

Finally, the volume enclosed by the LoD2 thematic surfaces – and a check on the water tightness of the resulting envelope – were computed and stored as generic attribute, too. An excerpt of the city model resulting from the geometry processing step is shown in Figure 2. Please note that, for better visual reference, all figures shown in this document are from the same area and not the whole municipality.

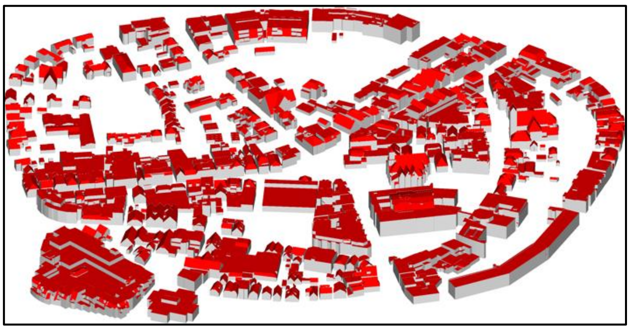


**Figure 2.** Excerpt of the 3D city model of Rijssen-Holten after the geometry processing step.

### 2.2 Dataset enrichment

Once the multi-part buildings were created, the geometries were checked and – if necessary – fixed as described before, the 3D city model was enriched by extracting and integrating data from other datasets. As primary source, the BAG dataset was chosen given its availability at national level (and its direct relation to the 3D BAG). The BAG id was used as the key to link the different datasets to the buildings. In the case of BAG, only circa 63% of the buildings could be classified according to their function (Figure 3, left).

Further analyses were carried out to better understand the distribution of buildings classified as “others” and “unknown”, according to their volumetric size. Table 1 shows the frequency and the aggregated volume of buildings per each class. While the numbers of “other” buildings are relatively small (e.g. the highest value is 315 out 22486, i.e. 1.4% for volume class 20-50 m$^3$), for “unknown” buildings the most frequent cases are found for relatively small volumes, i.e. 20-50 and 50-100 m$^3$. As a rule of thumb, one can divide the volume by an average height of 3 m

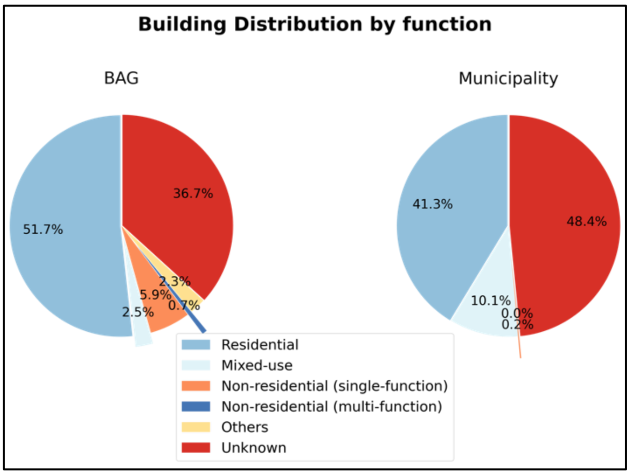


**Figure 3.** Distribution of buildings in Rijssen-Holten according to their function. Non-residential buildings have been further aggregated into single and multi-function. Data source: [left] BAG, [right] municipality or Rijssen-Holten.

and compute an equivalent footprint. For example, for a 100-m$^3$ building this corresponds to a footprint of circa 33.3 m$^2$, which can be safely assumed to be too small for a residential building. In other words, the largest part of the "unknown" buildings are most likely non-residential, small buildings (possibly garages, kiosks or depots, etc.). Such buildings would be anyway excluded from a simulation for heating energy demand purposes.

| Volume size range m$^3$ | Total |  | Others |  | Unknown |  |
|---|---|---|---|---|---|---|
|  | count | (%) | count | (%) | count | (%) |
| ≤ 20 | 447 | 1.99% | 16 | 0.07% | 430 | 1.91% |
| 20-50 | 2979 | 13.25% | 315 | 1.40% | 2662 | 11.84% |
| 50-100 | 2159 | 9.60% | 150 | 0.67% | 2007 | 8.93% |
| 100-200 | 1603 | 7.13% | 20 | 0.09% | 1399 | 6.22% |
| 200-500 | 8099 | 36.02% | 8 | 0.04% | 899 | 4.00% |
| 500-1000 | 5246 | 23.33% | 4 | 0.02% | 366 | 1.63% |
| 1000-2000 | 966 | 4.30% | 5 | 0.02% | 228 | 1.01% |
| 2000-5000 | 476 | 2.12% | 1 | 0.00% | 156 | 0.69% |
| 5000-10.000 | 262 | 1.17% | 2 | 0.01% | 67 | 0.30% |
| 10.000-20000 | 149 | 0.66% |  |  | 34 | 0.15% |
| > 20000 | 100 | 0.44% |  |  | 13 | 0.06% |

**Table 1.** Distribution of buildings classified as "others" or "unknown" according to their volumetric size.

| Volume size range m$^3$ | Total |  | Others |  | Unknown |  |
|---|---|---|---|---|---|---|
|  | m$^3$ | (%) | m$^3$ | (%) | m$^3$ | (%) |
| ≤ 20 | 7858 | 0.04% | 202 | 0.00% | 7068 | 0.04% |
| 20-50 | 104217 | 0.54% | 13377 | 0.07% | 88653 | 0.46% |
| 50-100 | 153656 | 0.80% | 9453 | 0.05% | 141421 | 0.74% |
| 100-200 | 234405 | 1.22% | 2868 | 0.01% | 196153 | 1.02% |
| 200-500 | 3033287 | 15.82% | 2638 | 0.01% | 277280 | 1.45% |
| 500-1000 | 3483349 | 18.17% | 2910 | 0.02% | 252395 | 1.32% |
| 1000-2000 | 1326705 | 6.92% | 5781 | 0.03% | 310174 | 1.62% |
| 2000-5000 | 1513085 | 7.89% | 2680 | 0.01% | 483330 | 2.52% |
| 5000-10.000 | 1913387 | 9.98% | 15495 | 0.08% | 475586 | 2.48% |
| 10.000–20.000 | 2299957 | 12.00% |  |  | 461740 | 2.41% |
| > 20.000 | 5099290 | 26.60% |  |  | 517126 | 2.70% |

**Table 2.** Aggregated volume of "other" or "unknown" buildings according to their volumetric size.

In terms of aggregated volume, it can be seen from the corresponding columns in Table 2 that these classes have indeed a small contribution to the overall volume in the city. Therefore, further strengthening the assumption that they can be safely ignored in the simulation process. Figure 4 shows the boxplot graph of the volume of the buildings volume considering the BAG function. Due to the variability of the dataset it was necessary to scale the values using a logarithmic transformation, so that graphs are better readable. From the graph it can be seen that mixed-use, and non-residential, the latter show larger distributions than the other 2 classes. In particular, again, the boxplot of the unknown buildings is in the lower part of the y-axis (small-size buildings) and it's rather compact.

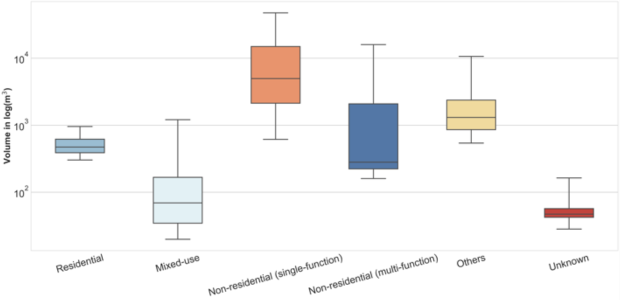


**Figure 4.** Distribution of gross volume of buildings in each function class according to BAG. Top and Bottom whiskers correspond to the 95th and 5th percentile, respectively.

In the case of the data coming from the municipality of Rijssen-Holten, a larger number of buildings could not be classified (circa 48%), as shown in Figure 3 (right). Additional work is being carried out to understand where the main differences are and to set up a set of rules to merge the two datasets and have a unique classification.

As a last step, the information about the building function was stored as CityGML building function attribute. The resulting CityGML-compliant dataset was validated using both FME and the 3DCityDB Importer/Exporter.

## 3. ENERGY SIMULATIONS

The 3D city model was used to perform energy simulation using two software packages. SimStadt, developed by HFT Stuttgart (SimStadt, 2021), accepts CityGML data as input and allows to perform different analyses on buildings, such as the computation of the energy-demand of buildings based on the energy-balance method (i.e. monthly and yearly values), estimation of solar irradiation and PV potential, etc. Besides csv format, SimStadt can also export CityGML data enriched with the Energy ADE, i.e. classes such as "EnergyDemand", "FloorArea", "RegularTimeSeries", etc. are used to store the results of the different simulations. This functionality is, at the moment of writing, still experimental, though.

CitySim Plus (CitySim, 2021) is an energy simulation software developed for dynamic simulation of clusters of buildings (based on a RC model). It is a software tool developed to provide urban energy planners with decision support using 3D geometrical buildings at urban district scale. Although the CitySim solver works with its own data model (CitySim XML file format), the GUI offered by CitySim Pro supports import of CityGML files, and export of CityGML with Energy ADE content, conceptually similarl to SimStadt. Unlike SimStadt, CitySim can generate results with a much finer temporal resolution, which can reach the hour.

In this section, a short report of the experiences with both simulation tools is presented. Besides running the software itself, an analysis of the data requirement was carried out each time, as well as an evaluation of the results (from a data structure and integrity point of view). Please note: the *accuracy* of the simulation results, as well as the *comparison* between SimStadt and CitySim are, at the moment, out of the scope of this work and is intended as a next step. The focus here is on understanding the requirements and setting up a workflow where a lossless data flow is guaranteed, including highlighting issues that currently hinder such a lossless workflow.

In order to reduce the simulation time in the testing phase, only a subset of the Rijssen-Holten 3D city model was used, consisting of 297 buildings (all single-part, except for a multi-part one), of which 79 have no information about the function according to the BAG. The CityGML building class and function values were additionally mapped to the corresponding numeric values contained in the codelists of CityGML 2.0 specifications (see Appendix C.1 of the CityGML standard). The codelists are actually only informative (therefore non-normative), and are based on the work of the German SIG 3D (SIG3D, 2021).

### 3.1 SimStadt

In this work, SimStadt 0.10.0 rev. 2021-05-06 was used. All simulations available in SimStadt are organized in workflows with several steps, depending on the type of desired analysis. Besides the 3D city model in CityGML (see later), climate information can be provided in four ways/formats: Photovoltaic Geographical Information System (PVGIS) (EU SCIENCE HUB, 2021), INSEL (INSEL, 2021), TMY3 Hourly and TMY3 Monthly. The first two are already available after the installation process. The latter two options consist of files to be uploaded. In the case of Rijssen-Holten, we used the PVGIS Database weather data source. Regarding the 3D city model, the CityGML files must contain buildings (both single-part and multi-part buildings are supported) that have been classified according to the function and for which the year of construction is known. The software first validates the input CityGML file (hence the importance of providing a valid CityGML instance document), then different operations are carried out depending on the user's choices.

We could successfully run the solar potential analysis (e.g. needed for photovoltaic potential), which requires the geometric data from the input CityGML file and a radiation model that could use the INSEL database available after the installation process. Figure 5 shows the isometric view resulting from the analysis. The export of the results is however not yet available as CityGML *with* Energy ADE, only as csv file. Post-processing is therefore needed to integrate the results with the input data.

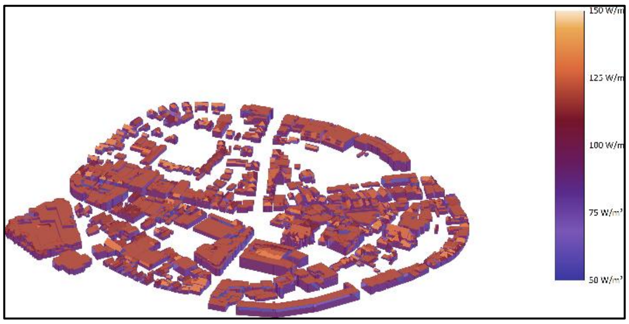


**Figure 5.** Screenshot taken from SimStadt showing the results of solar irradiance analysis carried out on a portion of the municipality of Rijssen-Holten in the Netherlands.

For heating demand estimation, the buildings contained in the CityGML input file must be characterised by means of physical properties needed for the simulation. Generally, these parameters (e.g. u-values, g-values, etc.) are included in libraries and assigned automatically depending on the building function and age class. SimStadt offers the possibility to retrieve these parameters automatically from a set of internal libraries. At the moment of writing, the available libraries are meant for Germany (based on the set of parameters by IWU – Institut für Wohnen und Umwelt), the Netherlands (based on TABULA) and the city of New York. From discussions with the SimStadt developers, it seems that the libraries are hard-coded, and, in any case, only a limited degree of customization is currently offered to the end-user. Besides, when trying to run the simulation on Rijssen-Holten using the Dutch library, problems were encountered: out of the 297 input buildings parts, only 48 were processed and passed the *PhysicsPreprocessor* step. As this seems to be a bug, the SimStadt developers were informed of it and it was decided, in the meanwhile, to proceed using the German library as a proxy one. In this case, all buildings could be processed and sent to the next step. Figure 6 shows a comparison of the results of using the Dutch (on the left) and the German (on the right) building physics libraries.

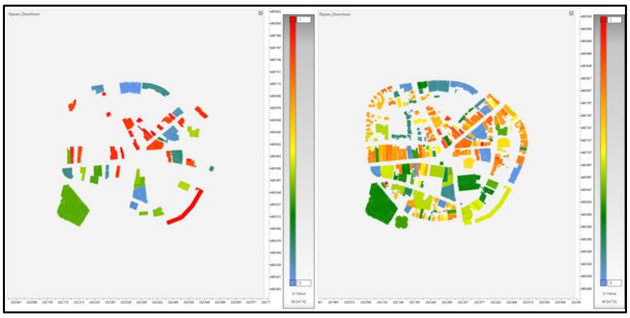


**Figure 6.** Comparison of the output of the *PhysicsProcessor* step using the different Dutch (left) and German (right) physic libraries.

| SIG 3D | | ALKIS | |
|---|---|---|---|
| **Code** | **Definition** | **Code** | **Definition** |
| 1000 | Residential building | 1010 | Residential |
| 1020 | Hostel | 2071 | Hotel |
| 1150 | Business building | 2050 | Retail |
| 1260 | Recreational site | 2081 | Restaurant |
| 1310 | Industrial building | 2111 | industry |
| 1970 | Administration building | 2020 | Office and administration |
| 2070 | Building for education and research | 3021 | Education |
| 2300 | Building for health care | 1022 | Health care |
| 2440 | Penitentiary or prison | 2060 | Hall |
| 2550 | Building for sport purposes | 3210 | Sport location |
| 2700 | others | 3036 | Event location |

**Table 3.** Building function codes translation table.

In the next step, the *UsagePreprocessor*, another hiccup was found. Despite apparently working in the previous step, the function attributes of the buildings must here be coded according to the codelists defined by the German ALKIS (AdV, 2021). Although a mapping could be carried out for most of the cases, sometimes some assumptions were required. Besides the usage of different numeric values for equivalent categories, in some cases there are multiple possibilities in the ALKIS codelists: for example, "education" can be mapped to ALKIS categories with codes 3021, 3022, 3065. In such cases, the first item in numerical order was chosen as the destination value. An example of the mapping table is given in Table 3.

The energy demand computation (for space heating and domestic hot water) was then started. For testing purposes, two simulations were run: the first using the building functions as they are, and the second setting all unknown buildings as residential. Figure 7 shows a screenshot of the results obtained from SimStadt for the heating energy demand. Figure 8 shows in a line plot the yearly values of heating energy demand for domestic hot water (blue line) and the space heating demand (orange line) of all buildings in the test area.

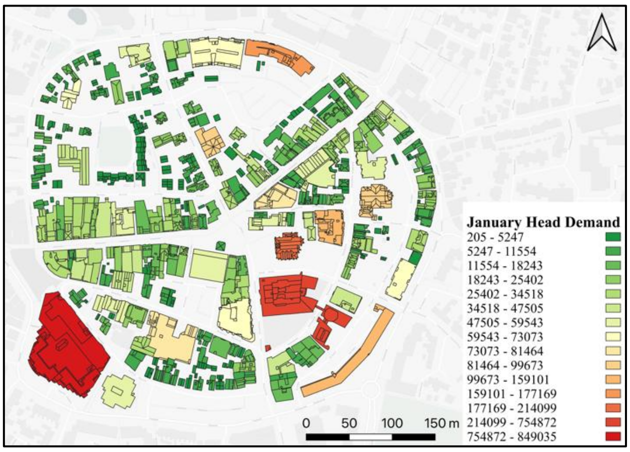


**Figure 7.** Heating energy demand results (for January) from SimStadt.

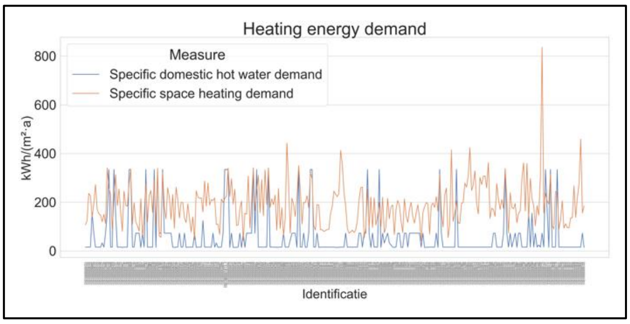
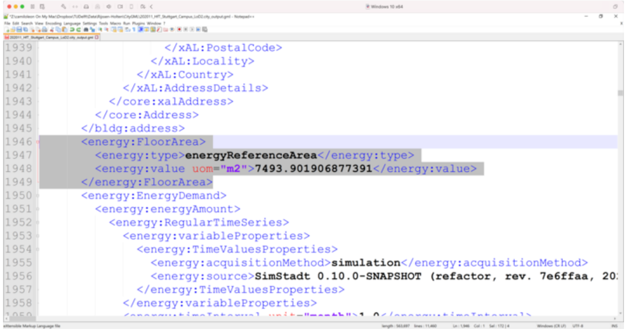


**Figure 8.** Line chart of the yearly heating energy demand results from SimStadt.

Results from the energy demand simulation can be exported also as a CityGML file (corresponding to the input file) with additional Energy ADE contents, i.e. the results of the simulation. More precisely, the Energy ADE contents follow the so-called KIT-profile of the Energy ADE, i.e. a subset thereof.

Before proceeding with the import of the resulting CityGML + Energy ADE file into the 3D City Database, a validity check was carried out on the output file. A number of issues were found, ranging from malformed XML encoding to wrong enumeration values. Examples of some of the cases are given in Figure 9: the CityGML (ADE) *_AbstractBuilding* property *floorArea* (mapped to the XML tag *energy:floorArea*) is missing, so that the *FloorArea* element is not correctly written. A similar issue applies to (ADE) _CityObject XML tag *energy:demands*, which should actually contain the *EnergyDemand* element. Another minor error was found in the value used as unit of measure for the property *timeInterval* of *RegularTimeSeries* class: "month" is used, but this not a valid value as per set of values defined in the corresponding enumeration (Ledoux, 2020). A workaround is to set it to 1/12 (i.e. 0.083) of a year. These errors have already been reported to the software developers of SimStadt. A workbench was created in FME so that is corrects the above-mentioned issues and writes (and checks for validity) a CityGML + Energy ADE (KIT profile) instance document.



```
                </xAL:PostalCode>
              </xAL:Locality>
            </xAL:Country>
          </xAL:AddressDetails>
        </core:xalAddress>
      </core:Address>
    </bldg:address>
    <energy:FloorArea>
      <energy:type>energyReferenceArea</energy:type>
      <energy:value uom="m2">7493.901906877391</energy:value>
    </energy:FloorArea>
    <energy:EnergyDemand>
      <energy:energyAmount>
        <energy:RegularTimeSeries>
          <energy:variableProperties>
            <energy:TimeValuesProperties>
              <energy:acquisitionMethod>simulation</energy:acquisitionMethod>
              <energy:source>SimStadt 0.10.0-SNAPSHOT (refactor, rev. 7e6ffaa, 20:
            </energy:TimeValuesProperties>
          </energy:variableProperties>
```

**Figure 9**. Sample output CityGML Energy ADE file from SimStadt

### 3.2 CitySim Pro

Analogously to the approach that was carried out in SimStadt, the same input dataset was tested in CitySim Pro. In this work, CitySim Pro 64-bit (Build 17 March 2021) was used. In terms of weather data, energy simulations in CitySim require a specific weather file format, named climate file ('.cli' extension), which offers hourly values throughout a year of records. Muntani *et al.* (2018) describe how a climate file is structured, how the header must be written, what the necessary fields and the data types for the values are. In the case of Rijssen-Holten, a Python script was developed to read and transform data from one of the closest weather stations (Twenthe) of the Royal Dutch Meteorological Institute (KNMI) into a suitable .cli file. Additionally, a horizon file is also required by CitySim, which was generated in order to take into consideration the surrounding orography of Rijssen-Holten (in general rather flat, however with some small hills nearby). For testing purposes and given the substantially flat orography of the test area, a simple horizon file was generated manually assuming for all directions an elevation angle of 0 degrees. Regarding the 3D city model, the CityGML file is read from the CitySim Pro GUI and converted to the CitySim's own XML data format. We could observe that:

- Multi-part buildings are not supported. This requires a prior restructuring of the original CityGML file so that, for example, the hierarchy is flattened and all building parts are transformed to buildings. Upon export, it must be however possible to reconstruct the original hierarchy, using, for example ancillary, temporary attributes;
- One building was identified and flagged as invalid, possibly due to geometrical issues that need to be further investigated. For the time being, this building was ignored and excluded.

Once the CityGML geometries were correctly imported, it was possible to perform simulations in CitySim, which runs nine simulations in just one click. Once the process is completed, the user may visualize in the GUI a desired field for a specific temporal resolution (hourly, daily, monthly and yearly values) and explore the results of the simulations interactively. Eventually, results can be exported also as CityGML + Energy ADE. Typically, these values are computed: short-wave irradiation (kWh/m²), long-wave net irradiation (kWh/m²), surface temperature (°C), photovoltaic production (Wh), solar thermal production (Wh), sky view factor [0,1], heating demand (kWh/m³), cooling demand (Wh/m³) and indoor temperature (°C). The temporal resolution can be chosen by the user.

If the user does not add specific building installations – such as PV panels –, no simulation is run for the photovoltaic production and solar thermal production. In this case, the user should add PV panel specifications either directly in the GUI or write their own CitySim xml file beforehand. For the other building characteristics, upon importing the geometries from CityGML, CitySim automatically assigns "standard" parameters from a pre-built library, which contains examples of boilers, heat pumps, PV panels, solar thermal, construction period, occupancy profiles, composites, etc. Therefore, this mechanism allows the software to run all the other simulations , e.g. heating demand, which requires U-values and g-values and that are not included in the original CityGML file. The drawback is however that it is not possible to assign automatically other customised parameters (e.g. tailored to the Dutch building stock), unless an external script that injects such information directly into the CitySim xml file is written. Alternatively, this operation can be done manually via the GUI, but it is obvious that it cannot be done for hundreds (or more) buildings at once.

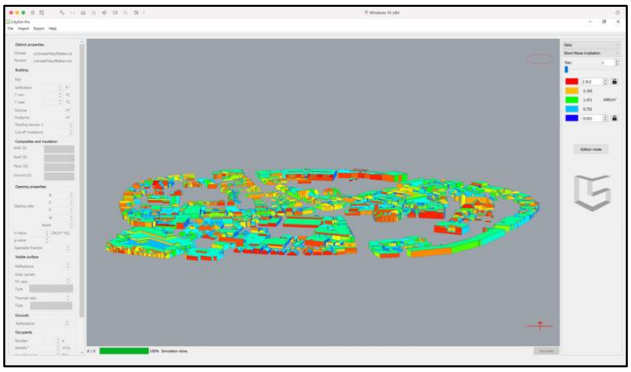


**Figure 10.** Example of visualisation in CitySim Pro of solar irradiation results.

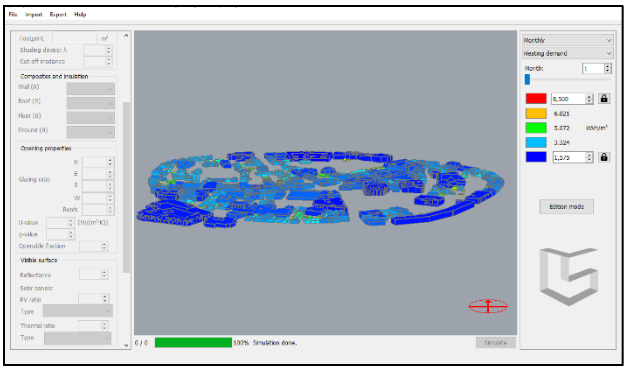


**Figure 11.** Visualising heating demand results in CitySim Pro. Default parameters (e.g. for g-values and u-values) are assigned to buildings and surfaces.

CitySim allows exporting the simulation results in several formats, including a CityGML 2.0 file which is then enriched with Energy ADE 1.0 contents. Analogously to the case of SimStadt, a validity test was carried out. Also in this case, a number of issues were found. First and foremost, it was observed that the output file is written including elements of the (full) Energy ADE, not only the KIT-profile. For example, information about PV panels (class *PhotoVoltaicSystem*) is contained, as well as several classes (e.g. *SingleValueSchedule*, *TimeSeriesSchedule*). While this is not a problem *per se*, it was noted that elements belonging to the *PhotoVoltaicSystem* class have invalid attributes (e.g. for azimuth and inclination), while other compulsory ones were left out (e.g. cellType). Additionally, some malformed XLinks were found.

As in the case of SimStadt, a (commented) workbench in FME was created to solve the issues and write a valid CityGML (+ Energy ADE) instance document. Additionally, certain data were adapted – or pruned, if necessary – to generate a KIT-profile compliant document. In the case of the *PhotoVoltaicSystem* objects, they were mapped to *GenericCityObjects*. Likewise, the developers of CitySim Pro were contacted to provide feedback and discuss the issues together.

## 4. DATABASE STORAGE

Once simulation results from both SimStadt and CitySim Pro were exported and stored as (valid) CityGML files with Energy ADE, the following step consisted in testing how to store these data into the 3D City Database. The 3D City Database (or, simply, 3DCityDB) is an open-source database implementation of the CityGML data model. It is available for both PostgreSQL/PostGIS and Oracle Database. Additionally, a number of tools are also shipped besides the database DDL scripts, such as an Importer/Exporter. For this purpose, two solutions were tested and will be briefly described in the following text. In both cases, the goal was to test whether a lossless flow of information is guaranteed, what the possible pit stops are, and which the solutions could be.

### 4.1 3DCityDB v. 4.3 with Energy ADE (KIT profile)

The 3DCityDB v. 4.3 (3DCityDB, 2021) was recently released (April 2021) and represents the latest update in the 4.x development branch. The most notable characteristic of the 4.x version is the added support for ADEs. This means that not only "standard" CityGML files can be imported and exported, but also files with ADE contents. ADE support depends on a number of modules/plugins that need to be available for the Importer/Exporter. They range from the possibility to add database objects (tables, stored procedures, constraints, etc.) to the 3DCityDB, to additional Java-based modules that extend the Importer/Exporter capabilities. Each ADE needs its specific set of the above-mentioned modules/plugins. In the case of the 3DCityDB 4.3, they are available as open-source software, although they only implement the Energy ADE KIT profile.

Nevertheless, starting from the files containing the simulation results, a number of bugs were encountered during the import (and successive export) process. All bugs were immediately reported and described to the 3DCityDB developers that quickly fixed them. At the time of writing (June 2021) they are solved and it is now possible to use the 3DCityDB 4.3 and its plugins for the Energy ADE (KIT profile) to import data into the database – and to export it again – without any data losses. The availability of such plugin greatly improves the interaction with the database as it automates the import/export operation and makes them accessible even via a simple, user friendly GUI that hides the underlying complexity of the database schema. The (current) drawback is the limitation to only the KIT profile, which somehow limits the full potential of the Energy ADE. For example, in order to store information about the PV panels from the CitySim simulation, they had to be converted to *GenericCityObjects*.

### 4.2 3DCityDB v. 3.3 with (full) Energy ADE

An alternative database solution to the 3DCityDB 4.3 is offered by an implementation of the (whole) Energy ADE v. 1.0 for the 3DCityDB 3.3. This is the result of an on-going in-house development within our group, and it builds upon the previous experiences with the 3DCityDB "Plus" (3DCityDB "Plus", 2017). Given the lack of ADE support in the 3.x release branch, the 3DCityDB "Plus" extended the last available version (3.3) and added ADE database support for a number of ADEs (Energy ADE 0.8, Utility Network 0.9, Scenario ADE 0.2).

On the one hand, the current version of 3DCityDB "Plus" adds support for the Energy ADE v. 1.0, and offers a number of additional database functionalities that are not otherwise available (updatable views, look-up tables, insert functions, etc.). Any Energy ADE data can be therefore stored, no matter whether it uses the full version, or only the KIT profile. On the other hand, there is no support for the Importer/Exporter, so that the user is faced with the task of directly interacting with the database structure. For example, this can happen programmatically (e.g. via Python scripts) or via FME workbenches that take care of converting data from CityGML. In our case, FME workbenches were created to transform and load ADE data from the CityGML files to the 3DCityDB "Plus" tables. The current version of the 3DCityDB "Plus" is planned to be released as open source in summer 2021. In a second moment, portions of it are planned to be ported to the 3DCityDB v. 4.3, in order to complement what is already available for that version.

## 5. CONCLUSIONS AND FURTHER WORK

The initial findings show that the proposed workflow is possible, i.e. it is possible to use the 3D BAG for energy simulations using SimStadt and CitySim, however there are still some major issues to overcome before the full pipeline can be carried out smoothly, without data losses, and deliver accurate results. Problems arise already at input data level when using the 3D BAG v. 2.0 dataset, and when trying to enrich it with data from other sources. Additional problems have been encountered when feeding the input 3D city model into both simulation tools. Software bugs at application level, limitations in the usability of the software (e.g. difficulty of using/defining building physics libraries for the Dutch building stock), problems in the format and validity of the resulting datasets have been found and documented. Also at database level some issues have been identified in terms of data import functionalities.

For the majority of the issues, some workarounds have been found and described in the paper, for others some additional tools or pre/post-processing tools are needed in the near future. Last but not least, it is hoped that the software developers of the respective simulation tools will improve their software and solve the reported bugs. This would reduce the number of workarounds implemented so far. A general note regards the need for better documentation of the simulation tools. Despite direct and fruitful contact with the software developers of SimStadt and CitySim Pro, there are still a number of undocumented (or poorly documented) features that raise the chances of generating wrong input datasets – or use the software improperly.

Among the several planned next steps, three are the main areas in which we are planning to conduct further work and that we would like to mention here:

- Better, consistent characterisation of the building stock. Although Rijssen-Holten has been chosen as a test case study, the ultimate goal is to extend the methodology to the whole Netherlands. Still, several in-between steps will be required to enrich the datasets sufficiently to be used for simulation purposes. For example, we are planning to carry out some tests using Machine-Learning approaches to better estimate the number of floors of buildings.
- Implementation of data-preparation tools that allow, for example, to automatically retrieve building parameters from (customised) libraries, in order to perform more accurate simulations. This applies in particular to CitySim, but it needs to be addressed in SimStadt, as well.
- Inclusion of other features in the simulation process. Examples are the current lack of a terrain surface, and of other possibly shadow-casting objects like trees.

## ACKNOWLEDGEMENTS

The authors would like to heartily thank Eric Duminil and Volker Coors (HFT Stuttgart), Jérôme Kämpf (Idiap) and Claus Nagel (Virtual City Systems) for their support with the respective software tools and the fruitful discussions.

## REFERENCES

AdV, 2021. ALKIS, http://www.adv-online.de/Products/Real-Estate-Cadastre/ALKIS/ (last accessed on 12 June 2021)

3D BAG, 2021, https://3dbag.nl (last accessed on 12 June 2021)

3DCityDB, 2021, https://github.com/3dcitydb (last accessed on 12 June 2021)

3DCityDB "Plus", 2017, https://github.com/gioagu/3dcitydb_ade (last accessed on 12 June 2021)

Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018, The Energy Application Domain Extension for CityGML: Enhancing interoperability for urban energy simulations. *Open Geospatial Data, Software and Standards*, 2018 3:2. SpringerOpen, United Kingdom.

CitySim, 2021, http://www.kaemco.ch/download.php (last accessed on 12 June 2021)

Doan, T.Q., León Sánchez, C., Peters, R., Agugiaro, G., Stoter, J., 2021, Volume comparison of automatically reconstructed multi-LoD building models for urban planning applications. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, ISPRS Congress 2021 digital edition (in press).

Dukai, B., Ledoux, H., Stoter, J, 2019. A multi-height LoD1 model of all buildings in the Netherlands. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, IV-4/W8, pp. 51–57.

Gröger, G., Kolbe, T., Nagel, C., & Häfele, K.-H. 2012. OGC City Geography Markup Language (CityGML) Encoding Standard. OGC 12-019. Open Geospatial Consortium. https://doi.org/OGC 12-019

Ledoux, H., 2020 Documentation for fieldgml https://3d.bk.tudelft.nl/hledoux/fieldgml/doc/withgml/temporal.xsd.html (last accessed on 12 June 2021)

Mutani, G., Coccolo, S., Kaempf, J., Bilardo M., 2018, CitySim Guide: Urban Energy Modelling. Create Space Independent Publishing Platform, ISBN-10:1987609735, ISBN-13:978-1987609738.

Praktijkhandleiding BAG, 2021, https://imbag.github.io/praktijkhandleiding/artikelen/welk-gebruiksdoel-moet-worden-geregistreerd (last accessed on 12 June 2021)

SIG3D, 2021, https://www.sig3d.org (last accessed on 12 June 2021)

SimStadt, 2021, https://simstadt.hft-stuttgart.de (last accessed on 12 June 2021)

Stoter, J., Peters, R., Commandeur, T., Dukai, B., Kumar, K., Ledoux, H., 2020. Automated reconstruction of 3D input data for noise simulation. *Computers, Environment and Urban Systems*, 80, 2020, pp. 101424.

Safe Software, 2021. FME. https://www.safe.com (last accessed on 12 June 2021)