

CITY ATB: CITYGML ANALYSIS TOOLBOX FOR ENERGY PERFORMANCE SIMULATIONS

A. Malhotra¹, S. Raming¹, J. Frisch¹ and C. van Treeck¹

¹Institute of Energy Efficiency and Sustainable Building (e3D), RWTH Aachen University, Aachen, Germany

ABSTRACT

District level energy simulations often require large datasets to extensively incorporate the semantic and topological information into the simulation environments. Moreover, the extraction and analysis of the desired data model can highly reduce the computational time resulting into higher efficiency. In this paper, an analysis toolbox is introduced which can be used to search, analyse and validate large CityGML datasets for district level energy simulations. Furthermore, the functionality to save the results into multiple industry friendly data formats is also further discussed.

INTRODUCTION

Emerging technologies, computational algorithms and simulation environments enable users, facility managers and occupants to achieve a good estimation of the energy demands of their buildings, districts and cities. This is a huge step forward towards the realization of the 7th United Nations Sustainable Goal of ensuring an affordable, reliable, sustainable and modern energy for the population of our planet (Economic and Social Council 2019). From the planning phases of a building envelop to the refurbishment phase of a fully constructed district, the ease and flexibility of searching, analysing and thereafter validating the appropriate data models for energy simulations is still limited. Although some public and private institutions do facilitate this process of data analysis, it is limited to a very small number of people and industries. Despite the deployment and implementation of a large number of software tools for energy simulations, this functionality to find and input the appropriate data model requires higher consideration. To bridge the gap between the recognition of an individual available building model, its analysis and validation, a framework is developed to enhance the user capabilities for a thorough preliminary investigation. Hence, human and computational efforts as well as costs relating to energy performance simulations at a building and/or urban scale can be highly reduced. Within the framework, an input functionality of City Geographical Markup Language (CityGML) data models for multiple levels of detail (LoD) with or without Energy ADE has been implemented. An

analysis for a large number of buildings and their building parts, along with validation algorithms, conforming to standardized schemas of the data models, from a building level to an urban scale is developed. This is integrated with the flexibility of saving the results in multiple industry friendly data formats. Therefore, by reducing the time and effort for the analysis of complex data models, the framework presented in the paper visions an increase in the adaptability of the CityGML standard between research communities, simulation scientists and urban planners. In the scope of this paper, open source CityGML data for the city of Aachen, Germany (Landesbetrieb Information und Technik Nordrhein-Westfalen, Ref 324 * Geoinformationszentrum 2017) is considered for further demonstrations. The implementation of the CityATB was carried out using PyQt (PyQt 2012) bindings for Python 3.5 and above.




CITYGML AND APPLICATION DOMAIN EXTENSION

City Geographical Markup Language (CityGML), an open Urban Information Modelling standard, is based on a modular XML schema structure. CityGML complies to the ISO 191xx family, the Open Geospatial Consortium (OGC), the W3C Consortium, the Web 3D Consortium and the OASIS standards. With 13 thematic extensions modules, the geometrical, semantic, appearance and topographical information of city objects representing cities, districts and individual buildings can be stored and exchanged using CityGML modelling standard (Gröger, et al. 2012). Depending upon the amount of information present in individual model, five Levels of Detail (LoD), as shown in *Table 1*, can be defined for CityGML.

The **Application Domain Extension (ADE)**, extends the core CityGML data models for multiple implementations (Gröger, et al. 2012). For energy performance simulations at city or district level, the building data models can be extended using the **Energy Application Domain Extension (Energy ADE)** (Agugiaro, et al. 2018). Similar to the modular structured CityGML data models, the Energy ADE extends the core city objects with the building physics

information, occupant schedules, material and construction, energy systems and with other information such as weather data.

Table 5: Overview of the five Levels of Detail for CityGML. Information retrieved from (Malhotra, et al. 2019)

Overview of LoD concept in CityGML (LoD0 – LoD4)	
	<p>LOD0 (2.5D model definition)</p> <ul style="list-style-type: none"> Regional and landscape scale representation Lowest accuracy No information over building installations Information about roof representations available No information about city furniture
	<p>LOD1</p> <ul style="list-style-type: none"> City and region scale representation 5/5m 3D point accuracy (Low) Object blocks as generalised features (>6*6m/3m) Flat roof structures No information over building installations and city furniture
	<p>LOD2</p> <ul style="list-style-type: none"> City, districts and project scale representation 2/2m 3D point accuracy (Middle) Objects as generalised features (>4*4m/2m) Differentiated roof representations Information over building installations available City furniture as prototypes, generalized objects
	<p>LoD3</p> <ul style="list-style-type: none"> City districts, exterior architectural models, landmark scale representation 0.5/0.5m 3D point accuracy (High) Object as real features (>2*2m/1m) Building installations as exterior features Real object form for roof structures and city furniture
	<p>LOD4</p> <ul style="list-style-type: none"> Interior architectural models, landmark representation 0.2/0.2m 3D point accuracy (Very High) Constructive elements and openings are represented Real object forms for building installations, roof structures and city furniture

To increase the use of CityGML data models for energy simulations and other multi-domain applications, a toolbox (*CityATB: CityGML Analysis Toolbox*) is developed. This tool will enhance the user's flexibility to pre-process and disseminate 3D models from a city to an individual building level. In the next sections, a detailed functionality of the *CityATB* will be discussed.

CITYATB: CITYGML ANALYSIS TOOLBOX

The *CityATB*'s user friendly architecture is structured according to different functionalities which can be used depending upon the application. A brief overview and further definitions of the different modules is made in the sections below.

For demonstration purposes in this paper, firstly an analysis of the open source data for the city region of Aachen is made. This functionality gives an overview of the number of buildings, along with some more attributes, and is further discussed in the section

analysis. Once the analysis is carried out, a search algorithm on the basis of coordinates and building usage is done. Furthermore, the desired area is exported as a new CityGML dataset and is again analysed. After analysis, the dataset is validated according to the CityGML standard schema. Furthermore, a time comparison with respect to the number of buildings is made for the different functionalities.

CityATB Main Window

The *CityATB*'s main window allows the users to navigate through the different modules and functions of the toolbox. With the independent implementation of individual functionalities and structure, the users of the tool will have a much larger flexibility in order to use CityGML datasets for wide range of applications.

Searching the appropriate buildings

Generally, CityGML datasets are a coagulation of tens of thousands of buildings clumped together in order to represent a large section as a city-quarter or the complete city itself. Moreover, it is often quite difficult to extract a small or large number of appropriate buildings for analysis and simulation purposes. This problem however results in higher computational effort and costs. The search module allows the user to narrow down the corresponding dataset. This can be carried out by entering a specific value, such as address or building functions and/or the coordinates of the area in which the building(s) are intended to be present. The search module of the *CityATB* allows the user to input the coordinates in different coordinate reference systems (CRS) using the EPSG Geodetic Parameter Dataset (IOGP Geomatics Committee 2020). Therefore, transformations within individual CRS are also possible.

Within the scope of this paper, open source CityGML LoD2 datasets for the city of Aachen, Germany from (Landesbetrieb Information und Technik Nordrhein-Westfalen, Ref 324 * Geoinformationszentrum 2017) are considered. The large dataset for the city region of Aachen contained 840 individual CityGML files which have 332,159 buildings and a size of approximately 6 GB in total. The analysis of the dataset was done using the Analysis module of the *CityATB* which is described in detail in the next section. A preliminary analysis gave the number of buildings inside each individual file along with their versions, coordinate reference system and the levels of detail.

For demonstration of the search algorithm, two techniques were used. Firstly, for searching the appropriate buildings or area, geographical coordinates from a web mapping service (in this case Google Maps (Google Maps 2020)) were taken and individually entered in the tool. The longitudes and latitudes of each geographical point are translated automatically inside the tool into the vertices of a

polygon. With an automated algorithm to convert the coordinates from Google maps to the respective CityGML file, a polygon is algorithmically created within the toolbox. Furthermore, an initial sorting of coordinates is carried out by computing the centroid of the area and then listing the coordinates by polar angle. The algorithm then compares the entered order with the sorted order for e.g. if the user enters 4 points P1, P2, P3, P4. The algorithm sorts and compares the two lists: if the two lists have the same order (or reversed order) and a different “first” point, such as: P2, P3, P4, P1, the programme will just continue; else the programme will ask the user if they would like to use the suggested or the entered order. For creating a closed polygon, a minimum of three vertices are required. Once the polygon is created, the search can be carried out to extract the desired buildings/area.

Figure 1 gives a visual representation of the buildings searched and outputted using the geographical



Figure 1: Visualization of the output file from the search algorithm. This will be referred as D1 for further demonstrations.

coordinates. This resulting dataset (further referred as D1) will be used for functional demonstrations within this paper. For visualizations of 3D city models, the authors used the FZK Viewer (IAI 2019) to input and visualize CityGML data models.

Using the second approach, different buildings on the basis of their usage/function/type were extracted. This

functionality could also be used with attributes such as addresses, number of storeys, etc. Within the CityGML standard, the definition of the attributes such as building usage, roof types and many more are made in separate code-lists. The code space element of the desired attribute identifies the code list through a globally unique URL. (Gröger, et al. 2012) Therefore, firstly buildings (as shown in Figure 2) with the modelled usage of a Tenement (CityGML Building usage codelist codespace: 31001_1010), from the larger dataset D1 were segregated into a new file.



Figure 2: Visualization of the buildings modelled as tenement within the dataset D1

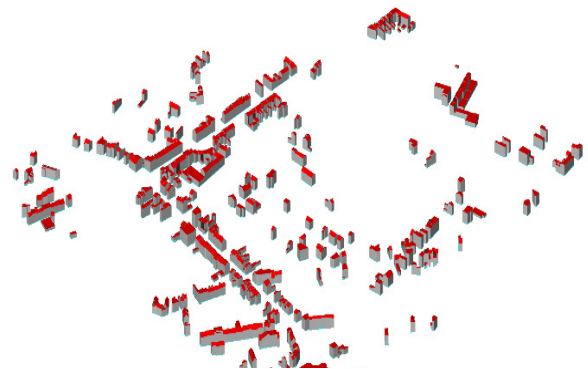


Figure 3: Visualization of the buildings modelled as offices within the dataset D1.

Table 2: An overview of the number of buildings and building parts modelled as offices or tenements in the output dataset of the coordinate search.

Dataset type	Codelist Codespace	Buildings	Building Parts	File Size (MB)
D1 (output of coordinate search)	-	2404	2798	72
Tenement Buildings	1010	838	1244	27
Office Buildings	1120	380	565	12
Residential	1000	1	0	0.13
Others	-	1185	989	-

Moreover, further segregation of the buildings (as shown in Figure 3) modelled as office buildings (codelist codespace: 31001_1120) was made. Once the offices and tenement buildings were located, new CityGML datasets, containing only the searched buildings were then outputted. This potentially, could save a lot of time and effort for the urban planners to analyse cities based on the building types. Table 2 gives an overview of the number of buildings and building parts modelled as Tenement or Offices in the target dataset D1.

Another important functionality of the CityATB is to scale the output area of a new dataset. As an example, if dataset D2 (as shown in Figure 4 (Left)) was searched using latitudes and longitudes, the desired building(s) are located and the geometrical area, using the geographical coordinates, is calculated. The user will have the flexibility to extend the original area by a factor between 1-100%. Once the incremental factor of the area is given by the user (for demonstration an increase of 50% was made), the algorithm will locate the buildings within the increased area and this could be outputted into a new CityGML file. A minimum input of three geographical points is necessary to search and extend the desired area. Figure 4 demonstrates a small extracted area in the city region Aachen. The buildings on the left were located using the geographical coordinates and the buildings on the right were outputted using the scaling algorithm. The area was scaled to increase by 50% compared to the original size.

The scaling for a new CityGML dataset was implemented using the Gauss theorem of area calculation of a polygon with known vertices (Meister 1769). According to which the area of a polygon can be calculated using (eq. 1),

$$A = \frac{1}{2} * \left| \sum_{i=1}^{n-1} (x_i y_{i+1}) + x_n y_1 - \sum_{i=1}^{n-1} (x_{i+1} y_i) - x_1 y_n \right| \tag{eq. 1}$$

where,

- A is the area of the polygon,
- n is the number of sides of the polygon
- $x_i, y_i, i = 1, 2, \dots, n$ are the ordered vertices of the polygon

which can be further simplified to (eq. 2),

$$A = \frac{1}{2} * |(x_1 y_2 + x_2 y_3 + \dots + x_{n-1} y_n) + x_n y_1 - (x_2 y_1 + x_3 y_2 + \dots + x_n y_{n-1}) - x_1 y_n| \tag{eq.2}$$

The scalability feature of the CityATB will allow the experts of multiple domains to use CityGML data models for many different applications. Obtaining the desired building(s)/area and further extending the data model with a domain specific extensions (ADE) will largely enhance the usability and accessibility in the planning and simulation phases. Some of the examples of such applications could be flood analysis, traffic, noise, air quality simulations.

Analysis

Independent to the different functionalities of the CityATB, large datasets can be analyzed before/after using any of the other functions. Using the analysis function, large data sets are discretely defined in a tabular form on the basis of the filenames, number of CityObjects (these are the elements that constitutes a CityObjectGroup in CityGML and are modelled using the Composite Designed Pattern from software engineering (Gamma, et al. 1995)), number of Buildings (geometrical models containing the details of individual built or unbuilt structures, represented as a Building feature) and BuildingParts (small underlying sections of a complex dismantled building structure; is modelled if the building is not homogenous or sometimes due to administrative

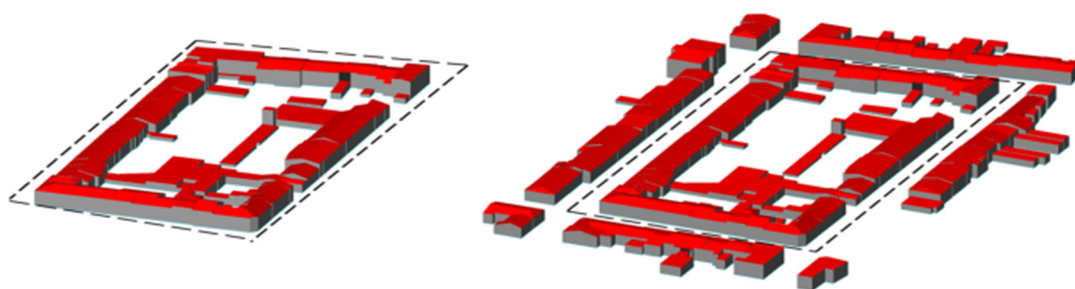


Figure 4: Visualization of the output dataset using the scaled function.

Left: The actual area for which the coordinates were inputted.

Right: Using the scaling function, visualization of a new dataset scaled up by 50% from the original dataset.

aspects), the levels of detail (LoD) (as explained in the CityGML section) in which the file is available, CityGML version (current developments are based on version 1.0 and 2.0), the coordinate reference system (used to geo-reference individual geometry elements) and file size (in Megabytes (MB)) in which the dataset has been defined.

Figure 5 gives an example overview of the analysis output from the dataset of Aachen. The results of the analysis can be stored for further usage into different formats. This is discussed in the Saving Results section.

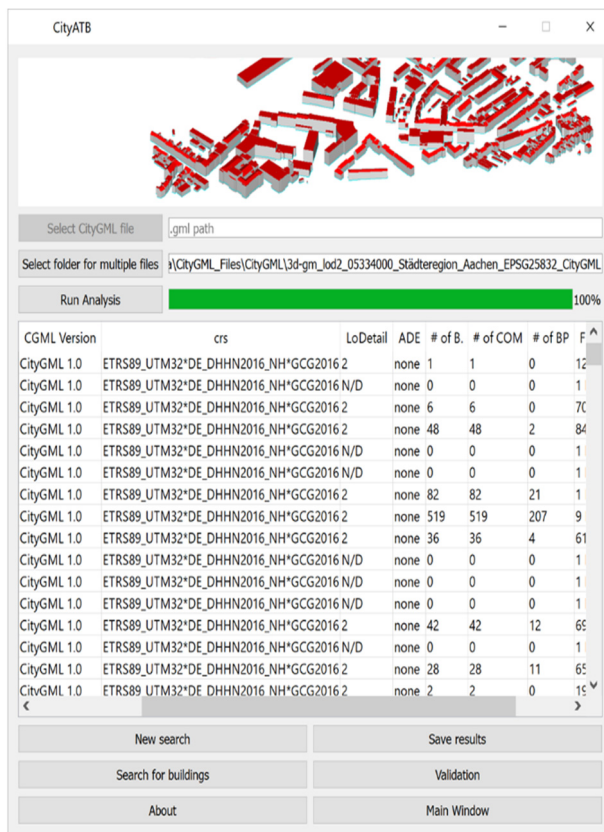


Figure 5: Example analysis result for the open source

The analysis of the open-source data for the city region of Aachen gave a detailed insight of the amount of buildings individual the dataset contains. Out of 840 datasets, 147 datasets did not contain any buildings or information. This number represents 17,5% of the total datasets and more importantly could potentially consume a lot of computational time and effort when inputting for urban scale simulations. Figure 6 gives a distribution of buildings in individual datasets.

Presently, the analysis function can be used for a single dataset or a folder containing multiple datasets and in future will also support the parsing functionality of the compressed ZIP files.

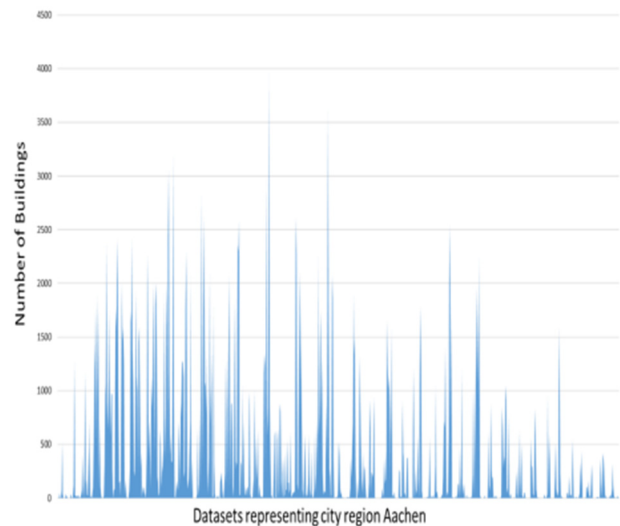


Figure 6: Distribution of the number of buildings per dataset

Validation

The functionality of the CityATB is based on the schema specific validation of the CityGML datasets. Generally, CityGML datasets (originated from the XML format) can be validated against the standards specified in the XML schema definitions (XSDs). Therefore, the validation of the user selected dataset is carried out using the inbuilt core CityGML schemas, currently implemented for versions 1.0 and 2.0. Furthermore, buildings or datasets which are extended using the Energy ADE and contain energy specific data can be validated against the CityGML 2.0 Standard along with the Energy ADE schema. Consequently, datasets can also be validated against user selected schemas. An json example output of the extracted file is shown below.

```
{ "CGML-ATB validation": {
  "LoD2_288_5631_1_NW.xml": {
    "schemaName": "CityGML1.0.xsd",
    "valid": "Valid!",
    "error": "None"},
  "LoD2_297_5630_1_NW.xml": {
    "schemaName": "CityGML1.0.xsd",
    "valid": "Valid!",
    "error": "None"}}}
```

An example of the validation output saved in json format.

A tabular form output within the toolbox window displays the details of the validation files along with the validation results. Furthermore, the detailed error description (as shown above) of the invalid files can be found in the exported results (explained in the next section). The analysis function can again be used to output different attributes of the valid dataset.

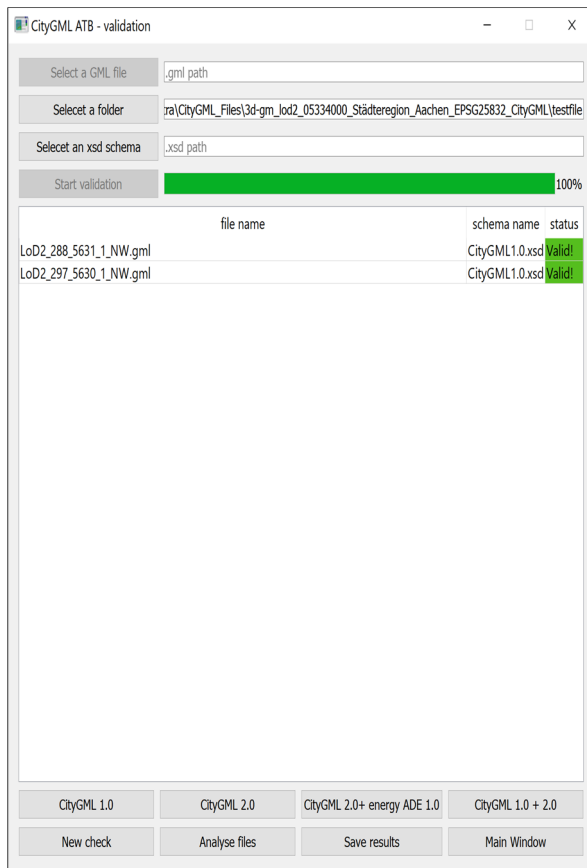


Figure 7: An example of the Validation Window

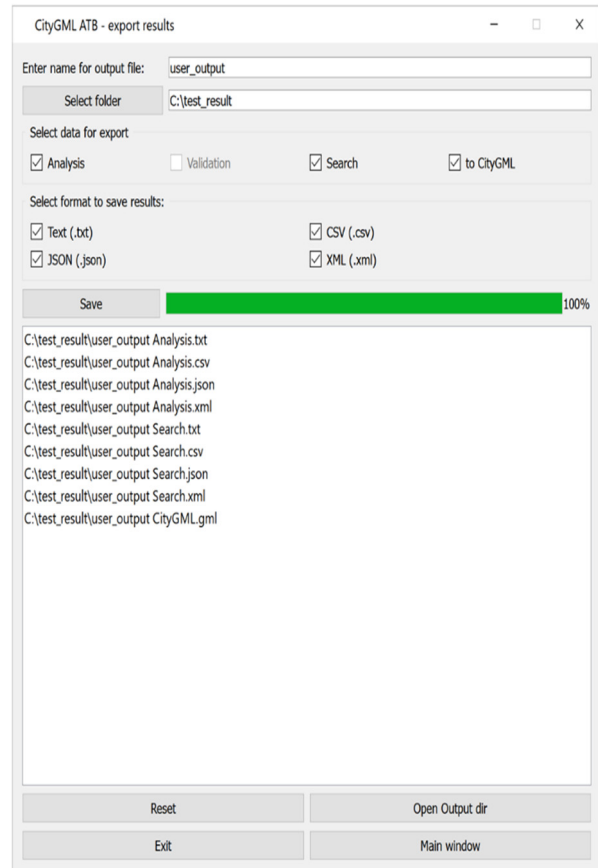


Figure 8: Window showing the saving of results.

Saving Results

Using CityATB, the users can save the results of each of the functionalities (search, analysis, validation) into multiple industry friendly formats.

Table 3, gives an overview of the input and output data formats that are supported by CityATB. In order to save the results from different functions, the user needs to select the functions along with the format in which the output data should be stored.

Table 3: Information about the input and out data formats supported by CityATB

Input	Output
CityGML	Comma separated value files (.csv)
XML	JavaScript object notation (.json)
	Extensible mark-up language (.xml)
	Text files (.txt)
	CityGML (v 1.0 and 2.0)

Although the name and directory of the file to be saved is set to pre-default, it can be selected and changed by the user. Once the selection of the output results and the format in which the results are intended to be saved is made by the user, the output file path is shown in the window.

The motivation behind the support of multiple data formats of the CityATB is to enhance the user’s flexibility to further use the data for energy- or simulation-based applications. Governmental organization and many industrial partners mainly use “.csv”, “.txt” and “.xml” formats, whereas, research institutions and simulation scientists are more inclined towards using “.json” and “.gml” datasets. In future, the authors would like to add the functionality of plotting and saving the analysis results into different picture formats (such as jpeg, png). Therefore, this feature will help a wide range of implementers to analyze and use the exported data.

Time Comparison

In order to optimize the accessibility and functionality of using large CityGML datasets, computational time calculation is one of the important key performance indicator. Moreover, the trade-off between scalability and efficiency on one hand and portability and transformability on the other is one of the core challenges in the scientific community (Pflüger, et al. 2016). High performance computing along with multi-core processing is the main focus of the community in order to optimize the time effectiveness in different applications (Wyngaard, et al. 2013).

As CityATB is developed with the focus of optimizing the time and effort for pre-processing large datasets in urban scale energy applications, time comparison graphs for the different functionalities highlighting the computational time with respect to the number of buildings is shown below. For demonstration of the execution time (in Figure 9, 11 and 12), dataset D1 was considered as the reference area. Furthermore, along with D1, the datasets extracted using the search algorithm (as shown in Figure 2 and 3) were considered to compare the execution times. One of the datasets consisted of buildings modelled as offices (380 buildings) and the other dataset consisted of buildings modelled as tenement (838 buildings). Moreover, comparison for computational times (in Figure 10) for the building(s)/area searched using geographical coordinates was also made. Using the open source data of the city region Aachen, out of 332,159 buildings, smaller areas with 42, 635, 1564 and 3515 buildings were extracted.

These smaller areas that were selected to compare the computation times were extracted using the CityATB's (geographical polygon) search module. For searching 42 buildings (using 5 coordinate points), the algorithm took 392.59 seconds. Whereas, for 635 buildings (using 12 coordinate points), 1564 buildings (20 coordinate points) and 3515 buildings (using 35 coordinate points) the searching time was 393.26 seconds, 393.58 seconds and 413.92 seconds respectively. However, comparing to the value search the computational time for coordinate search was higher. This is due to the pre-processing of the coordinates into the respective reference systems and thereafter searching of the appropriate buildings in the user defined area. Moreover, the analysis and validation of the buildings within the dataset D1, showed an approximate linear increase in the time with respect to the number of buildings. In future, the authors would also like to implement and test the algorithms using multiprocessing computations.



Figure 9: Graph showing the amount of time taken for searching of building using the value search.

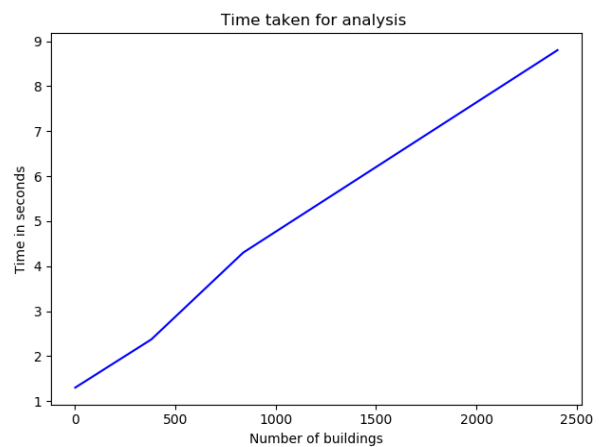


Figure 11: Graph showing the amount of time taken for analyzing different buildings.

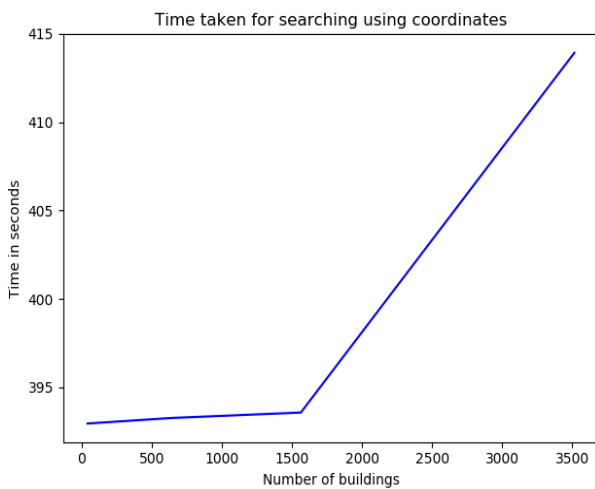


Figure 10: Graph showing the amount of time taken for searching different buildings with geographical coordinates.

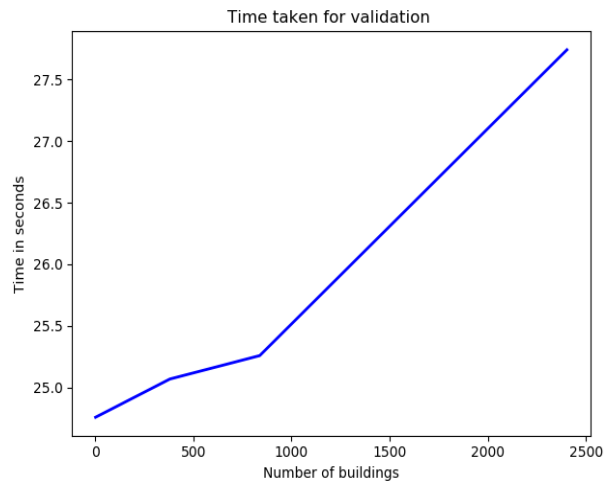


Figure 12: Graph showing the amount of time taken for validating different buildings.

CONCLUSION

Urban scale energy simulations are often quite complex and do require a large amount of data imports and exports. In order to improve the efficiency of our simulation tools, extraction and transformation of the appropriate data model is very important. With CityATB, the authors were able to accomplish the desired data model segregation which further helped and drastically reduced the importing, computational and exporting time for urban scale energy simulations using Modelica (Mattsson and Elmqvist 1997). Nevertheless, the simulations could also be facilitated by many other simulation environments and tools such as EnergyPlus (NREL 2019) or TRNSYS (Klein, et al. 2017) which support the input functionality of CityGML data models (Geiger, et al. 2019). Due to data protection laws in different countries, the authors restrict to open the information of the exact coordinates or addresses of the datasets used. Therefore, random coordinates were chosen for using the open source data models. Moreover, due to the user-friendly architecture and provision of detailed documentation of the tool, the authors believe that this toolbox could be used by beginners and expert users alike. Furthermore, this tool, which will be available as open source (<https://gitlab.e3d.rwth-aachen.de/e3d-software-tools/cityatb>), and can be extended for further requirements and applications. With the possibility to adapt different requirements of the users, as the next steps, developments such as automatic conversion of CityGML versions, geometrical transformations, datasets merge function integration are envisioned by the authors. Moreover, a web-based functionality to import and export open source data models is also foreseen.

ACKNOWLEDGEMENT

The authors would like to thank all the partners of the Energy ADE and IBPSA Project1 consortium for the detailed and helpful discussions.

REFERENCES

- Agugiaro, G., J. Benner, P. Cipriano, und R. Nouvel. „The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations.“ *Open Geospatial Data, Software and Standards*, 5. March 2018.
- Economic and Social Council. *Special edition: progress towards the Sustainable*. United Nations, 2019.
- Gamma, E., R. Helm, R. Johnson, und J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- Geiger, A., J. Benner, K. H. Häfele, und V. Hagenmeyer. „Building Energy Simulations at Urban Scale Based on Standardised Data Models.“ *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*. Rome: IBPSA, 2019.
- Google Maps. *Aachen, Germany*. 2020. <https://www.google.de/maps/place/Aachen/@50.7596743,5.9564408,11z/data=d50.7753455!4d6.0838868> (Zugriff am 03. February 2020).
- Gröger, G., T.H. Kolbe, C. Nagel, und K.H. Häfele. *OGC City Geography Markup Language (CityGML) Encoding Standard*. OGC, 2012.
- IAI, KIT. *FZKViewer (5.2)*. 2019. March 2019. <https://www.iai.kit.edu/1302.php> (Zugriff am 19. February 2020).
- IOGP Geomatics Committee . *GeoRepository*. 2020. <https://epsg.org/home.html> (Zugriff am 02. 02 2020).
- Klein, S.A., Beckman WA, Mitchell JW, und et al. *TRNSYS 18: A Transient System Simulation Program*. Solar Energy Laboratory, University of Wisconsin, Madison, USA, 2017.
- Landesbetrieb Information und Technik Nordrhein-Westfalen, Ref 324 * Geoinformationszentrum. <https://www.opengeodata.nrw.de>. 2017. https://www.opengeodata.nrw.de/produkte/geobasis/3d-gm/3d-gm_lod2/ (Zugriff am 30. January 2020).
- Malhotra, A., M. Shamovich, J. Frisch, und C. van Treeck. „Parametric Study of different Levels of Detail of CityGML and Energy ADE Information for Energy Performance Simulations.“ *IBPSA Building Simulation*. Rome, 2019.
- Mattsson, S.E., und H. Elmqvist. „Modelica - An international effort to design the next generation modeling language.“ *7th IFAC Symposium on Computer Aided Control Systems Design*. Ghent, Belgium: CACSD'97, 1997.
- Meister, A. L. F. „Generalia de genesi figurarum planarum et inde pendentibus earum affectionibus.“ *Nov. Com. Gött. (in Latin)*, 1769: 1-144.
- NREL. *EnergyPlus*. 27. March 2019. <https://energyplus.net/> (Zugriff am 21. July 2019).
- Pflüger, D., et al. „The Scalability-Efficiency/Maintainability-Portability Trade-Off in Simulation Software Engineering: Examples and a Preliminary Systematic Literature Review.“ *Fourth International Workshop*. Salt Lake City: SE-HPCCSE, 2016. 26-34.
- PyQt. „PyQt Referencing Guide.“ 2012.
- Wyngaard, J., M. Inngs, J. Collins, und B. Farrimond. „Towards a many-core architecture for HPC.“ *23rd International Conference on Field programmable Logic and Applications*. Portugal: IEEE , 2013.