FLEXIBLE FILTERING OF HETEROGENEOUS DATA USING THE EXAMPLE OF THE DESIGN AND SIMULATION OF BUILDING INTEGRATED PHOTOVOLTAICS

J. Eisenlohr[1], C. Eller[2], U. Rüppel[2], C. Leifgen[2], M. Boudhaim[1], C. Maurer[1], W. Sprenger[3] and T. E. Kuhn[1]

[1] Fraunhofer Institute for Solar Energy Systems ISE, Freiburg, Germany
[2] University, TU Darmstadt, Darmstadt, Germany
[3] Ed. Züblin AG, Stuttgart, Germany

ABSTRACT

According to the Building Information Modeling (BIM) methodology, all participants must be provided with the information relevant to their tasks in the required level of detail at the appropriate time. In the future, the amount of detailed digital information will further increase. Therefore, we propose rule-based filtering beyond the purely class-based extraction of subsets of the overall model. The source of digital information can be a central data model, from digital product data of component manufacturers or generally from the worldwide web. After the rule-based filtering, automatic parsing into the target language takes place. As an application example, we demonstrate this methodology for the simulation of building-integrated photovoltaics as an example.

Introduction

Due to increasing digitalization in the planning processes and the operation of construction projects and buildings, amounts of data, formats and requirements for software systems are increasing. A continuous and consistent data storage over the entire building´s life cycle is challenging, especially considering the large number of participants involved. Nevertheless, all planners require specific information at a specific time with a certain level of detail to fulfil their own tasks. An example for a complex planning and construction process is the holistic integration of energetically active facade components like building integrated photovoltaics (BIPV). Reliable simulations can only be carried out on the basis of a consistent data exchange with sufficiently detailed data sets. A concept to match these requirements to the heterogeneous data sets is proposed in this paper: The actual data structure should not be prescribed, but flexible filters developed that can extract the desired information with the required level of detail from the existing data.

Related Work

There are no solutions to automatically sort, edit and provide different information in different levels of detail depending on the use case. The current solution's approach prescribes that rules can be used to filter information from data, if it has a specific factual basis or structure. Here it is possible to automatically filter out data in different formats or levels of detail (Yoon et al. 2017). This data can only be automatically analyzed or pre-processed by the prescribed data storage or structure. Such a structure is described by Zhang (Zhang et al. 2018) or VDI3805 (VDI3805). In a heterogeneous data landscape, which is usually the case in BIM, these structures are difficult or impossible to setup due to the large amount of data. Zhang et al. showed an approach for data selection and processing in BIM processes in the field of energy technology by implementing a combination of different techniques (Zhang et al. 2018). By means of preparing data and machine learning, they obtain information for calculating the energy balance automatically. They use data from different sources, but these are available in a specified file scheme. Machine learning, however, can only be successfully carried out with the help of very specific structures and purposeful data. A versatile, inhomogeneous and very variable data basis such as in BIM is very challenging for the application of machine learning. For the use case presented in this paper, the integration of photovoltaic systems in buildings, the required data is very heterogenous and thus not well suited for machine learning.

Another concept for filtering of building data in a BIM process is the IDM/MVD method (buildingSMART) proposed by buildingSMART. By this method, IFC files are created using filters. The defined schema version is used to verify the integrity of IFC files. The result of filtering by the IDM/MVD method is an IFC file generated according to predefined specifications. In this case, the rules are schemas predefined by the IDM. The effort involved in defining the individual data requirements for individual specialist roles via the IDMs is very time-consuming and only slightly flexible. It also works only with building data in an IFC-File, which is not the whole data in a Building Information Model like is shown in the design and simulation of building integrated photovoltaics. The advantage of the IDM/MVD method is to define processes precisely and to create filters that can be used permanently for these processes. However, as processes in different construction projects are usually not equal or not using the same information, the method is not usable. There is a multitude of product and building information data in very different formats. Nevertheless, the information of one area is also needed and linked to the information of another one. An exclusive method for the individual construction planning as described above is therefore not desirable.

As the planning of building integrated photovoltaic systems requires a large variety of different information from different sources, we look at this application example to demonstrate the versatility and flexibility of the approach proposed in this paper. Photovoltaic systems that are integrated into buildings (BIPV) face several challenges compared to free-standing PV plants. Besides their multi-functionality and a more complex construction process with the constraints given by a building and its facades and roof, BIPV systems are often geometrically very complex, leading to a more complicated electric circuit and effects like partial shading and hot spots. While for plant or rooftop configurations, simplified models at module level can be applied, geometrically and electrically complex BIPV systems require tools that are fully flexible with regard to the geometrical and electrical design of the system and a description at solar cell level. This makes planning processes and yield analyses of BIPV systems complex and time-consuming. A complete BIPV simulation requires not only an electrical model of the PV (empirical models like (Heydenreich et al., 2008) or (King et al., 2004) or equivalent circuit models like (Chin et al., 2015)), but also prior steps like irradiance calculations and thermal modelling of the PV devices and subsequent steps like DC-AC conversion. Especially the calculation of irradiance can be challenging, as irradiance data of high spatial resolution on each solar cell of the BIPV system are required. As the geometry of the building skin and its surroundings is usually complex, simple view factor methods widely used for simple PV plant configurations cannot be used. Detailed, time-resolved shading analysis is required. There are several tools on the market that can fulfill parts of these requirements, mainly 3D CAD programs like for example ArchiCAD, GoogleSketchup, Autodesk Revit or specialized PV simulation tools like for example PVsyst, PV*Sol or BIMsolar. However, all of these tools have certain restrictions and none accurately combines all relevant BIPV simulation steps: detailed irradiance calculation (1), temperature calculations of PV cells (2), electrical DC behavior of individual cells (3) and the complete system (4) and DC-AC conversion (5). Therefore, BIPV planning processes are performed in very different ways with different level of detail and varying quality by different planners.

CONCEPT

Due to the numerous demands on data at different points in the planning processes, as shown in the related work, it is important to find a solution that enables flexible data processing and filtering without restricting the amount and structure of data. Therefore, a concept was developed that suggests a flexible way to prepare the required data in different levels of detail, independent of the available data, data structure and amount of data.

To extract specific content from a set of information, procedures must be defined that tell the processing machine what to do. This can be done in various ways. In the simplest case, clear rules of the form "If premise, then conclusion" are defined. In order to map more complex relationships, decision tables can be created similar to simple rules, in which one or more actions can be defined for combinations of any number of conditions. In more complex applications, artificial intelligence (AI) methods can be used instead of fixed rules, but these are not addressed here, because they can be implemented for a specific application but cannot be used for a broad spectrum. By the introduced approach, we propose a solution, which combines simple filtering approaches with the power of more flexible routines.

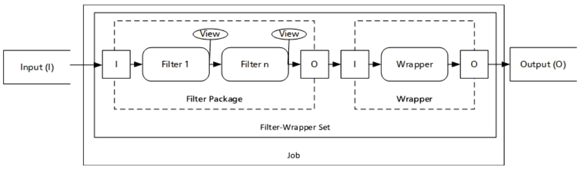

Figure 1: Dataflow in a filter-wrapper set

The concept is based on a filter-wrapper set (Figure 1). Such sets can be assembled and developed according to individual requirements. A set has the task of filtering input data, preparing it, for example, merging or redesigning it and finally wrapping it into the desired output data format. Filtering is done by a various number of individual filters, which together form a filter package. All filters have a defined input and output. Only one data source can be processed per filter and the data format of the output corresponds to that of the input. The simplest form consists of a filter package with one empty filter, in which the input and output are equivalent. This is necessary, for example, if data sets are to be queried completely. Before and after each filter, the results can be looked up and examined in a view like the views processed in query languages like SQL. Views can also be output and used for checks during the filter process.

To be able to use data in different applications, it may have to be converted from one format to another. This can be done by means of so-called wrappers for the respective requirement. Such a wrapper receives the data in a specific input file format and converts it into the desired output file format so that this can be used by a specific software or another connected interface. For example, the building data of an IFC file can be converted into a RADIANCE file for ray tracing calculations. The simplest form of a wrapper has the same input as output format and no conversion takes place.

In order to further increase flexibility, an additional module can be added to the entire preparation process. This corresponds to a worker, which has an arbitrary number of inputs and outputs (while a filter and a wrapper only have one input and one output). This allows for using information from one input to process another input. As an example, translating a file into another language can be mentioned. Besides the file that shall be translated, a file that contains the desired language as information is needed. The worker uses both information sources to change or add language data based on the language selection. In this case, a worker parses the data into the desired output format and no wrapper is needed. This powerful component makes a filter-wrapper set very flexible and adaptable (Figure 2).

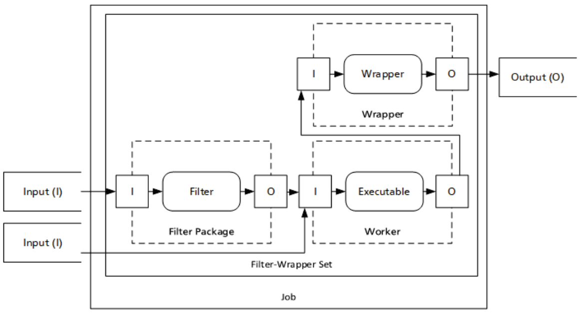

Figure 2: Dataflow in a filter-wrapper set with multiple inputs

The idea behind this concept is to develop an adaptable environment that provides sets of filters and wrappers for various applications and data formats. Once these sets have been defined, they can be re-used in a later project phase or in further projects. With the help of this concept, there are no general limits to the pre-processing of data to required information, but only to the individual requirements and implementation of the individual filters, workers and wrappers.

IMPLEMENTATION

To implement the concept, a MEAN (mean) (MongoDB, Express Framework, AngularJS (Frontend) and Node.JS (Server)) stack was used. MEAN enables the implementation of various interfaces. For example, a REST API offers the possibility to connect to any other system. MEAN stacks are asynchronous systems that are fast and scalable. Furthermore, the development is beneficial for developers through a simple setup and building. MongoDB as NoSQL database offers a greater flexibility compared to conventional relational DB systems. One reason for this is that all content is stored in JSON format. This is also used natively in Node.JS, so the data can be used directly and no parsing or schema migration is necessary.

The server provides a backend, which is responsible for creating, viewing and executing the filter-wrapper sets. These can be carried out in so-called jobs via a browser-based frontend or the API (backend) via external software solutions, e.g. Autodesk Revit with input files from the cloud or the world wide web. The Node.JS server stores the jobs in lists so that they can be executed separately from the MEAN stack by a Java application. This application permanently runs in the background and essentially comprises a thread that executes all jobs waiting in the database one after the other. The jobs are assigned different states (waiting, in execution, successful, failed). The results in the form of the outputs are stored on the server and can be downloaded and used by planners in a parsed data structure and file format via the API or the frontend.

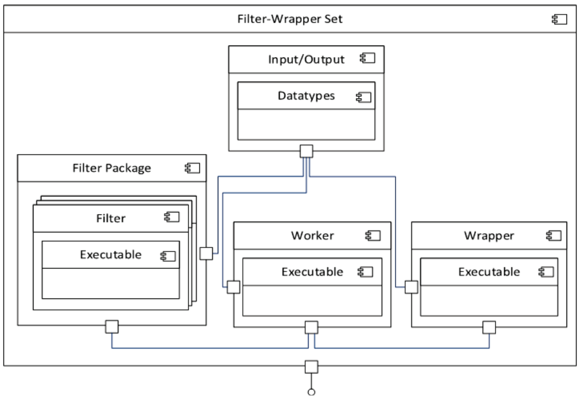

Figure 3: Components and relationship of the filter-wrapper sets

On the server, the blocks described in the concept are added as individual components and linked to the MongoDB of the MEAN stack (Figure 3). As mentioned before, the input and output component have the knowledge about the file format and schema if necessary. This can be used to do sanity checks in advance in the future. This component is required for all other components of the filter-wrapper set. Filters, wrappers and workers are able to process a wide variety of data in a variety of ways. The applications and scripts required for this can be correspondingly diverse. To make this possible, when creating the filters, wrappers and workers, an application to be executed, the so-called executables, is specified with any parameters in such a way that the command entered must only be executable in a Unix based shell. Thus, different programming languages and frameworks can be used. For example, Java Archive (JAR) or Python applications are used as executables. This demonstrates the technical feasibility of flexible filter wrapper sets.

USE CASE: BIPV SIMULATION

As described in the related work, the planning of a BIPV system requires various information from manifold sources. Today, a planner has to collect nearly all this information manually and add it in his specialized planning tool.

Fraunhofer ISE has developed a very detailed and specialized planning and simulation tool for complex BIPV systems (Chin et al., 2015; Sprenger et al., 2016; Eisenlohr et al., 2017). Collecting the required input for this tool suite is time-consuming and error-prone. Therefore, the BIPV simulation tool suite is used here to demonstrate the versatility and applicability of the developed filtering concept.

The objective of the BIPV simulation tool suite described in the following is to calculate the detailed electrical and thermal behavior of all solar cells of a BIPV system at each time step. The tool suite comprises five major parts that are depicted in

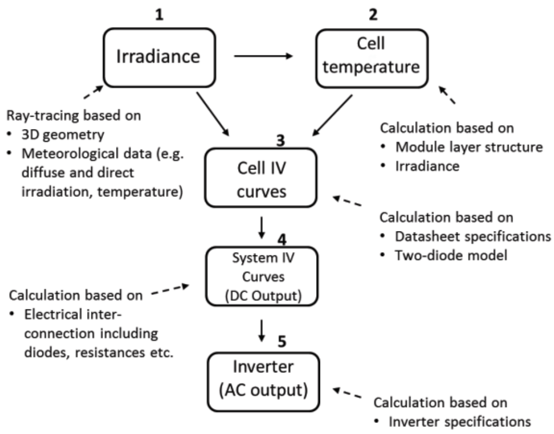

Figure 4: Schematic description of the five main steps for a detailed BIPV simulation. For each time step, the irradiance on each solar cell, the temperature of each solar cell, the cell IV characteristic for each temperature/irradiance combination, the system IV

The following Table 1 summarizes the most important required input for all the 5 steps, shows where this information usually comes from and in which format it is typically available.

Table 1: Information required for a BIPV-planning and simulation with the corresponding sources and typical data formats.

Step	Information	Source	Format
1	3D geometry	CAD programs	eg. .rvt, .3ds, .skp, .pln, .dxf, .dwg, .ifc, .obj, …
	meteorological data	meteorological databases	.pdf, .txt, .xls, .csv
2	Module layer structure	Module data sheet	.pdf
3	Electrical cell parameters	Module data sheet	.pdf
4	Interconnection	PV-planner	
5	Inverter Specifications	Inverter data sheet	.pdf

Obviously, there are a lot of different data sources and formats involved. Typically, many of these formats are not fully machine readable (like pdf documents) and/or not standardized. The BIPV simulation tool suite takes text files and a geometry file in the RADIANCE formats .rad/.mat as input that have to be created based on the various sources. Following the concept presented above, we developed sets of filters/filterpackages/wrappers/workers that allow a more automatized collection of the required input data. Three important sets are given in Table 2 as examples. The first two, ifc2rad and ds2dm, are described in the following in more detail to demonstrate the conceptual ideas.

Table 2: Example sets of filters, wrappers and workers to extract, collect and pre-process data for a detailed BIPV simulation.

Step	Input	Filter/wrap-per/worker set	Output: Files usable for BIPV simulation tool suite as input
1	Model of complet e building as IFC file	ifc2rad	.rad and .mat containing the geometry and materials containing the building skin
3	Digital model of module	ds2dm	.txt containing only the relevant parameters for the 2-diode-model fitting
5	Digital model of inverter	inv2ss	.txt containing only the relevant parameters for the fitting of a parametric inverter model

Ifc2rad

The filter-wrapper set ifc2rad takes an IFC file as input and performs a combination of filtering and wrapping. It filters information that is not related to geometry or material definitions and objects that are not relevant for the irradiance calculation on the outer building skin. This comprises for example geometrical details from the interior. Reducing the amount of information is essential as the resulting file size correlates to the required calculation time in the ray tracing. This filtering currently still needs manual adjustments, but a higher degree of automatization is currently under investigation and would perfectly fit in the developed framework. Also an automated reduction of geometric details is a currently investigated step that can be included in the proposed concept. To demonstrate ifc2rad, Figure 5 shows a building that is currently planned at Fraunhofer ISE. On top an IFC file exported from a BIM-software (visualization with Solibri Model Viewer) and on the bottom a visualization of the RADIANCE geometry after executing ifc2rad. The trees for example have been filtered out, because they only exist in the architectural model but not in reality. In addition to filtering, ifc2rad translates the geometry from IFC, which supports several different geometrical representations like boundary representation or constructive solid geometry, to RADIANCE, which doesn't support all of these representations.

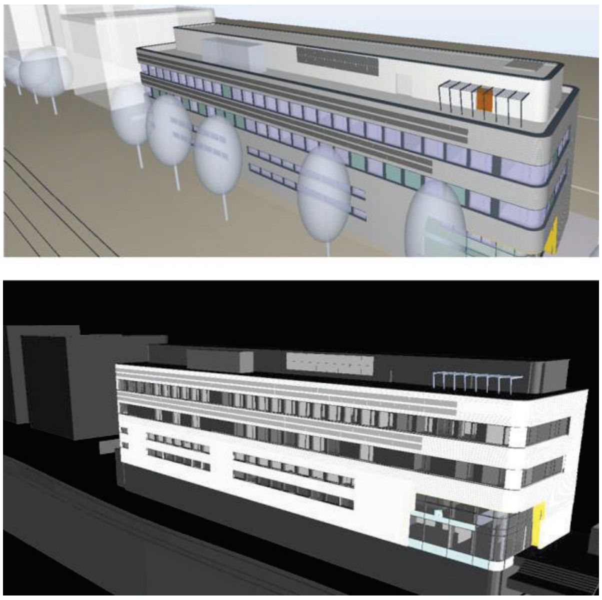

Figure 5: Visualization of an IFC file (top) of a building planned at Fraunhofer ISE. Visualization of the RADIANCE file (bottom) after application of ifc2rad, which comprises filtering and wrapping.

Ifc2rad is also a good example to demonstrate the re-usabilty of filter-wrapper routines. Once, such a modular set of filters and wrappers has been created, it can also be used for different applications. It generally allows for a link between BIM models, which can be exported to IFC, to applications that require ray-tracing. Day-lighting simulations for example have successfully been coupled with ifc2rad and can use the same or some parts of the filter-wrapper set.

<u>Ds2dm</u>

As part of the BIPV simulation, a two-diode model of each solar cell in the system is created. To determine the two-diode model parameters, characteristic electric properties of the PV module like open circuit voltage, short circuit current and maximum power point current and voltage are required. Therefore, the filter-wrapper set ds2dm extracts this information from the digital product model via queries. As product descriptions may differ (e.g. one manufacturer specifies the named current and voltage values, another manufacturer could specify a complete IV-curve that contains these current and voltage values), these queries have to be as flexible and "intelligent" as possible. Figure 6 shows a visualization of the filter-wrapper set ds2dm by filtering the required data for the fitting of the two-diode model. Input_1 is a TTL file that is provided by the digital product data base presented in (Wagner et al., 2018). The filterpackages FP_1 and FP_2 remove all information from the product data model, that are not relevant for the extraction of two-diode model parameters, e.g. the geometrical properties of the module or other attributes like weight, price etc. Wrapper 1 translates the remaining attributes from .ttl to .txt and worker 1 translates the output in the language defined by Input_2.

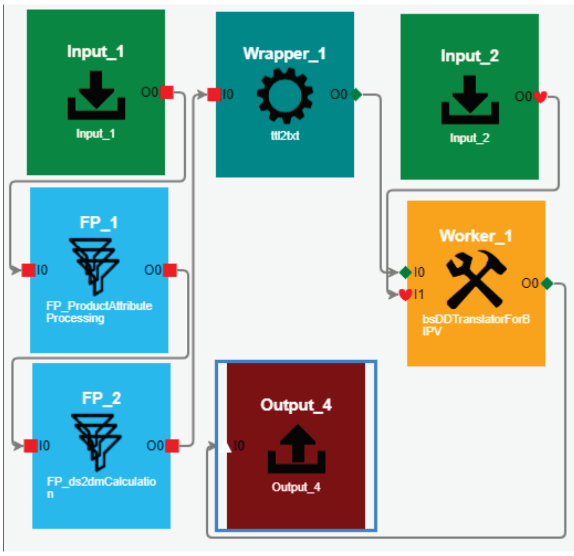

Figure 6: Visualization of the filter-wrapper set on the frontend to provide the required data for the fitting of the two-diode model.

CONCLUSION

In this paper, we presented a concept for flexible filtering of large amounts of heterogeneous data. In the BIM methodology, for a specific task certain information are required with a specific level of detail. As the amount of data in construction projects and thus in BIM models increases rapidly, efficient filtering and pre-processing is required to extract the data relevant for a specific task and to provide it in the required format and level of detail. Neither prescribing the format and structure of the source data strictly, nor machine learning, nor the IDM/MVD method from buildingSMART offers a general solution adapted to the heterogeneous and versatile data existing in a construction process and its planning phases. Therefore, we propose an adaptable environment containing filters, wrappers and workers that can be tailored to specific tasks, combined in different ways and re-used for related tasks and applications. We used a MEAN stack as basis for the implementation. The filter-wrapper sets are created, stored, managed and executed on a server in the backend, which is accessible via a browser-based frontend or the REST API for other software applications. For the filters and wrappers there are no fundamental or conceptual restrictions as long as they can be executed from a Unix based shell.

The necessity for such a filter-wrapper-framework and its flexibility and versatility have been demonstrated in an exemplary use case: planning and simulating a BIPV system. To design and optimize such a BIPV system, detailed simulations are required and these simulations require diverse input data from various sources. The required filter-wrapper sets for a BIPV simulation have been implemented and can provide input data for a detailed BIPV simulation in a much more automated way than before. For the planning and optimization of BIPV systems, which currently is time-consuming and highly non-standardized, the coupling to BIM methodology with such partially automated, “intelligent” and adaptable filter processes is a very promising approach.

REFERENCES

Wagner, A., Möller, L. K., Eller, C., Leifgen, C., Rüppel, U.: SolConPro: An Approach for the Holistic Integration of Multi-Functional Facade Components into Buildings' Life Cycles. Accepted for presentation at the 17th Internatrional Conference on Computing in Civil and Building Engineering, 05.-07.06.2018, Tampere, Finland.

buildingSMART. Online available at https://www.buildingsmart.de/bim-knowhow/standards.

Chin, V.; Salam, Z.; Ishaque, K. (2015): Cell modelling and model parameters estimation techniques for photovoltaic simulator application. A review. In: Applied Energy 154, S. 500–519. DOI: 10.1016/j.apenergy.2015.05.035.

Eisenlohr, J.; Wilson, H. R.; Kuhn, T. E. (2017): Detailed Modelling of Building-Integrated Photovoltaics - From Component and Environmental Data to the System Output. 5 pages / 33rd European Photovoltaic Solar Energy Conference and Exhibition; 2118-2122. DOI: 10.4229/EUPVSEC20172017-6DO.11.1.

Heydenreich, W.; Müller, B.; Reise, C. (2008): Describing the World with three Parameters: A new Approach to PV Module Power Modelling. In: 23rd European PV Solar Energy Conference and Exhibition (EU PVSEC), S. 2786–2789. DOI: 10.4229/23rdEUPVSEC2008-4DO.9.4.

King, D.L, Boyson, W.E., Kratochvilll, J.A (2004): Photovoltaic Array Performance Model. Report SAND2004-3535. In: Sandia National Laboratories (3535), S. 1–43.

mean. Online available at http://mean.io/.

Sprenger, W.; Wilson, H. R.; Kuhn, T. E. (2016): Electricity yield simulation for the building-integrated photovoltaic system installed in the main building roof of the Fraunhofer Institute for Solar Energy Systems ISE. In: Solar Energy 135, S. 633–643. DOI: 10.1016/j.solener.2016.06.037.

VDI3805. Online available at http://www.pit.de/deu/Produkte/pit%20-%20VDI3805/pit-vdi3805.htm.

Yoon, J.; Seo, W.; Coh, B.; Song, I.; Lee, J. (2017): Identifying product opportunities using collaborative filtering-based patent analysis. In: Computers & Industrial Engineering 107, S. 376–387. DOI: 10.1016/j.cie.2016.04.009.

Zhang, C.; Cao, L.; Romagnoli, A. (2018): On the feature engineering of building energy data mining. In: Sustainable Cities and Society 39, S. 508–518. DOI: 10.1016/j.scs.2018.02.016.