# Machine-code functions in BIM for cost-effective high-quality buildings


Christoph Maurer*[1], Wendelin Sprenger[2], Steffen Franz[3], Marouane Boudhaim[1], Jan Lodewijks[2], Uwe Rüppel[3], Tilmann E. Kuhn[1]

[1] Fraunhofer Institute for Solar Energy Systems ISE, Heidenhofstr. 2, 79110 Freiburg, Germany

[2] Ed. Züblin AG, Albstadtweg 3, 70567 Stuttgart, Germany

[3] Institute of Numerical Methods and Informatics in Civil Engineering, TU Darmstadt, 64287 Darmstadt, Germany

*corresponding author: Dr. Christoph Maurer, Fraunhofer ISE, Heidenhofstr. 2, 79110 Freiburg, Germany, phone +49 761 4588 5667, email christoph.maurer@ise.fraunhofer.de



Building Information Modeling (BIM) is one way of making the complex building processes less expensive and more reliable. This paper first analyses existing experience with building processes which include building-integrated solar systems as one example of buildings with high energy-saving goals. Based on the analysis, the authors propose to include a new property set and functions in machine code in the next version of the Industry Foundation Classes (IFC). For the machine-code functions, several formats are proposed and the advantages and disadvantages of the proposed IFC extensions are discussed. As several formats have specific advantages and disadvantages, it is recommended to offer them as an option. For simple conventional building components, the existing IFC version 4 seems appropriate. Buildings with innovations that add complexity to the building process can profit most by the savings generated by exchanging machine-code functions. Cost-effectiveness can be increased because more knowledge can be shared due to the confidentiality offered by the machine code, less time is needed for variations of the models, the higher accuracy of detailed models can be used, more competition is possible due to the modularity of the functions and fewer mistakes are made while exporting and importing models using the same validated model.



*Keywords: simulation; building information modelling (BIM); building-integrated solar systems (BISS); building-integrated solar thermal systems (BIST); solar building envelope; building process; industry foundation classes (IFC)*


## 1. Introduction

Scenarios for the energy transition such as [1,2] typically include a reduction of the energy demand of buildings as well as renewable energy sources. The European parliament and council has decided that from 2020 on, all new buildings have to be nearly zero-energy buildings (NZEB) [3]. The energy-relevant requirements for new and refurbished buildings are becoming more stringent, for example in Germany with [4–8]. At the same time, more and more “green buildings” fulfil the high requirements of certificates issued by LEED [9], BREAM [10] and DGNB [11]. Recent publications such as [12] document that building-integrated solar systems can reduce costs compared to refurbishing the building envelope first and adding solar systems later. It is therefore likely that solar building envelopes will become more and more important for the construction industry, especially regarding high-quality buildings that aim to achieve a zero energy balance or even a plus energy balance. With the additional function as a converter of solar energy, solar building envelopes are more complex than conventional building envelopes. The current paper analyses first the existing building process and experience with demonstration buildings including solar envelopes, and proposes a method to handle solar envelopes more cost-effectively. The building process extends in this case from the planning phase, through construction, up to the facility management. Finally, the advantages and disadvantages of the propositions are discussed and conclusions are drawn.

## 2. Theory

Many building processes to date involve several stakeholders, who receive the necessary input as text e.g. in an email or a 2D report. Then they apply their own methods and tools and share their results e.g. as another 2D report. This often functions, but much information is lost; some information has to be generated by more than one stakeholder and is not shared with other stakeholders who could save costs by using it. It is also a risk for the quality of the building process if only parts of the generated information are shared because different stakeholders may assume different values for the information that has been withheld. Errors can also occur when information is imported into a 2D report and exported from such a report. One reason for this may be the effort of sharing more information, the confidentiality of some trade secrets and the fear of being responsible for all shared information.

As the exchange of information can make the whole building process more cost-effective and reliable, there have been efforts by several companies within buildingSMART [13] to establish the Industry Foundation Classes (IFC) [14] as an open format for exchanging information during the building process. For example, the exchange of a 3D geometric model including semantic information can save a lot of effort. The building process is also addressed by buildingSMART [15,16]. However, the IFC scheme separates building components and elements of distribution grids. Combined components like solar building envelopes have not yet been introduced into the IFC and the processes. To illustrate some specific issues raised by solar building envelopes, two examples for the integration of an innovative transparent solar thermal collector into two buildings will be analysed.

During the development of a transparent solar thermal collector (TSTC) within the project described in [17], it became clear that the energy-relevant properties of a solar thermal façade like the g value (also known as "solar heat gain coefficient", "solar factor" and "total solar energy transmittance") depend on the operation mode of the façade collectors [18]. At the same time, the efficiency of the façade collectors also depends on the building services which use the renewably generated heat. At low temperatures, this efficiency is considerably higher than at high temperatures. To evaluate the new collector concept, a simulation model of the collector was developed [19]. As high-rise buildings were targeted, a typical floor plan was developed, which included offices facing each cardinal direction as well as a corridor, meeting room, photocopier room, kitchen and core areas. The same façade element, including a window with internal venetian blinds, an opaque part concealing the floor slab and the transparent collectors in the spandrel area, was used for all offices. The HVAC system included thermally activated building systems, a sorption chiller and a compression chiller, a gas boiler, buffer tanks and hybrid coolers. Different volumes of the heat storage tanks were investigated, as was the option to use the building mass for additional thermal storage.

Although it is just a case study, this evaluation is complex and hardly feasible for a single stakeholder. A large façade company contributed its expertise on façade elements and high-rise floor plans, an experienced HVAC planner designed the building services and a research institute modelled the transparent collector. The results were published in [20]. The researchers were not experienced in treating building services and the HVAC planner did not have experience with the detailed physical model of the façade collector. Therefore, the complexity of the collector modelled was encapsulated in a new Type in TRNSYS [21], which is easy to use, and the building services were modelled in TRNSYS and MATLAB [22]. To exchange a TRNSYS TypeX, a Dynamic Link Library DLL file is exchanged. It contains the function TypeX() in machine code that is used by the TRNSYS solver and a "proforma file" which is used only for the TRNSYS graphical user interface. In addition, an example deck for the collector model, the building and the building services can be provided to make the start easier. Based on the simulations, it would be possible to carry out a real high-rise building project.

Apart from this case study, a demonstration installation of transparent façade collectors was planned [23], built and monitored [24]. The building process is analysed here according to the process map of [25]. During the project development, suitable locations for the active solar area of the building envelope have to be identified first by at least the architect or building owner, together with a partner with experience in solar energy. In some cases, it is obvious that the upper part of the south façade is the most suitable position. In other cases, simple whole-year simulations can compare e.g. a west façade to a partially shaded south façade. An estimation of the energy demands of the building is essential to evaluate different building systems which use solar thermal energy. The solar (building) envelope should also fit well aesthetically into the design of the refurbished building.

During the technical design phase [26,27], a simulation model of this solar-thermal building envelope was developed by the research institute. This simulation model can be used for other applications of these collectors or could be modified quickly for similar collector variants. With the simulation model, the HVAC planner compared the maximum heating and cooling supply of the building services with the heating and cooling demand. It is essential that the specifications of the components that are assumed for this maximum heating and cooling supply and the functional description of the building services are met by the components which are later installed by another company. In parallel, the building site was inspected, at least by the façade manufacturer, in order to plan the refurbished building envelope in detail including the installation of the solar-thermal façade. In this case of a demonstration building, an extensive monitoring system is recommended. In a commercial application without scientific evaluation, a few sensors should be included to allow the facility manager later to check the performance of the building-integrated solar-thermal (BIST) system. If the system is changed at a late stage of the technical design, the preceding results should be updated. In this case, an additional flat-plate

collector field was installed as a back-up without changing the design of the building services which resulted in suboptimal performance.
During the planning of work, a timeline of the refurbishment was developed according to which the building was refurbished in the construction phase. During the facility management phase, the values of the sensors should be checked for plausibility as soon as the system is operating. In this case, with scientific monitoring, the monitoring values from several months should be analysed. In the case of a commercial installation, the system performance should be checked after one year of operation because then data is available for different seasons and operating conditions. If the system is not operating as expected, the controls or other components should be optimized.
Some of the necessary tasks can be performed by one stakeholder. For example, a solar consultant can analyse defined areas of the building envelope with his tools and recommend the most suitable area to the architect. For other tasks, it seems beneficial if the knowledge of one stakeholder is transferred to another stakeholder. For example, the HVAC planner is empowered to perform simulations of the building, the solar envelope and the building services by receiving an easy-to-use model of the solar envelope created by a research institute. The research institute or test lab could also measure and model BIST elements for a manufacturer of solar envelopes, who could then offer the simulation tool together with his BIST elements to clients. One option is to share the source code of the simulation model. This can be helpful in many cases. In the case presented above, the BIST manufacturer probably does not want his competitors to know the physical details of his solar envelope. A model which is shared in machine code can hide the details. Therefore it could be shared easily. A model could also be shared if all inputs, outputs and equations are described in a text file. This would disclose details of the component to competitors and it would be much more difficult to convert the model into text format and from text format into an executable simulation model than to share the model in machine code e.g. by sending a DLL file. Without a validated simulation model, innovative building-integrated solar systems (BISS) are difficult to sell because important advantages cannot be quantified. In addition, the innovative BISS components can hardly contribute with their advantages to the energy transformation without a building process that can handle solar envelopes.
This analysis was done for building-integrated solar thermal systems as an example, but the results are applicable also for other solar envelopes such as building-integrated photovoltaics (BIPV), building-integrated photovoltaic-thermal (BIPVT) systems and passive building envelopes including daylighting and solar control.

## 3. Results

An open format to exchange information during the building process - like the Industry Foundation Classes (IFC) - is important because actors in a building process may use software from different manufacturers and still need to exchange their information. If one software manufacturer dominated the market with a closed format, the software prices could rise. The current version of the IFC is version 4 [14]. The aim for the future is that all necessary input for more than one stakeholder can be exchanged in the IFC and that the stakeholder can return his output in the IFC format. However, several properties which may be needed for building processes of high-quality buildings are not yet available in IFC version 4. Table 1 presents some examples of these properties together with a short explanation of the context in which they may be needed. They could be added to the next IFC version e.g. as a property set Pset_BuildingEnvelopeDetails.

*Table 1 Examples of properties which may be needed for building processes of building envelopes*

| Property | Explanatory example |
|---|---|
| Distribution of the irradiance on the building envelope | For whole-year shading simulations |
| Wind direction | Important for unglazed BIST |
| Air pressure distribution on the building envelope | Important for rear-ventilated BISS |
| Angle-dependent polarization-dependent spectral transmittance | Result of optical measurements of individual layers |
| Angle-dependent polarization-dependent spectral reflectance | Result of optical measurements of individual layers |
| Angle-dependent absorptance of a layer including multiple reflections | Result of optical simulations of BISS elements |
| Angle-dependent transmittance of the building envelope including multiple reflections | Result of optical simulations of transparent BISS elements |
| Refractive index | Result for uncoated transparent materials |
| Solar and light extinction coefficient | Result for uncoated transparent materials |

| | |
|---|---|
| Longwave infrared transmittance | Input for thermal simulations |
| Number of positions of movable parts | For switchable and adaptive building envelopes |
| Characterisation of each position e.g. by its angle-dependent absorptance und transmittance | For switchable and adaptive building envelopes |
| Solar heat gain coefficient / g value depending on the irradiance and collector operation mode | Result of a BISS model |

Some properties of Table 1 could have the format of IfcTimeSeries or average single values for the entire year or distributions like a histogram or nested lists. If, for example, a sub-component of a BIST façade is to be specified with high accuracy – e.g. one individual glass pane for the BIST façade of a high-rise building – then the values for several angles, two polarizations and hundreds of different wavelengths should be exchanged between the glass manufacturer or the laboratory measuring this glass pane and the stakeholder who creates the simulation model of the BIST element. This can be costly and error-prone since there is currently no generally accepted format for such data for automated export and import of this information. Additional difficulties for the specification of such a format are that different formats are used by different software programs and that the requirements on the format depend on the component itself. As an example, it can be mentioned that the spatial resolution needs to be very high at least in some solid angles for façade elements with strongly angle-dependent properties and that the format has to be capable of variable spatial resolutions. However, even if there is great flexibility and complexity in the specification of the properties of complex façade elements, it is still possible that new innovative components are out of scope of the format and that they cannot be specified reliably with the given format. An alternative is to exchange functions in machine code for the description of the properties of the façade elements. If functions in machine code are allowed in future IFC versions, this would ensure that innovative components can always be described.

If we assume that functions in machine code have been exchanged and that a solar thermal façade has been constructed on this basis, then the analysis of the next BIST building process can illustrate the capabilities of this approach. Figure 1 presents a schematic drawing of five functions in machine code. Assuming that only the glass pane facing the exterior is changed in the BIST element, the function of the old glass pane can be easily replaced by the new function, and the optical properties of the entire BIST element as well as the energy simulation model of the BIST element can be generated immediately, because the optical properties of the other subcomponents and their thermal properties remain the same. Function 4 needs the definition of the building and the building services, which could be imported from IFC. Then Function 4 can provide the energy demand and comfort in the building. In addition to the output of Function 4, Function 5 also needs input in the form of data characterising the internal loads and the weather to calculate performance indicators to assist the facility manager. Function 5 uses Function 4, which uses Function 3 and so on.

Typically, different models are used at different levels of detail by different stakeholders although the same model could indeed be used. Much information and sometimes important characteristics are lost by using a much simpler model at the next higher level of aggregation. For example, a high-performance building envelope may be modelled in detail for the comparison of different variants, but for the building simulation, a much simpler model may be used and the building services may be designed with an even simpler model of the building. However, adaptive functions can be written which can switch between different levels of detail. The functions need to call their sub-functions only once if their results are not time-dependent. In many cases, the calculation times present the limiting factor. Nevertheless, functions in machine code can have short calculating times as long as they are programmed well.

On the one hand, this process is cost-effective because little time is needed to adapt the machine-code functions to a new building process. On the other hand, it offers great flexibility, which can also save costs. No single expert needs to write the complete BIST simulation model with one code, but different stakeholders can contribute with what they can do best and the results can be used by the other stakeholders to create machine-code functions for numerous tasks.

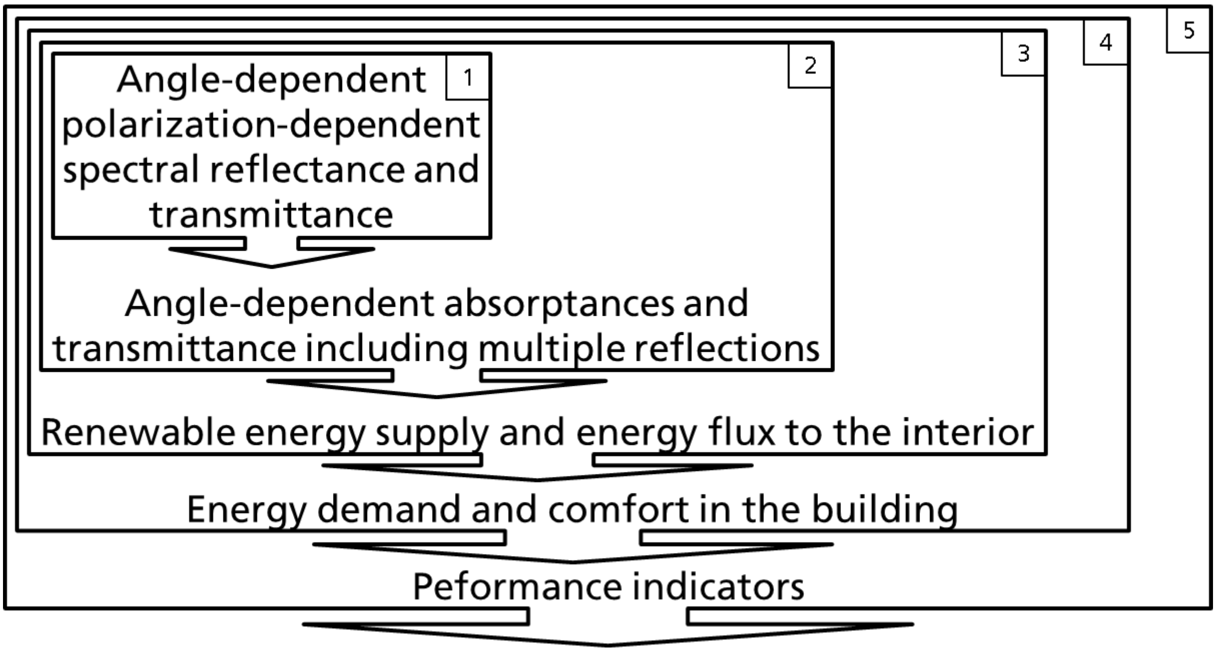


*Figure 1 Schematic drawing of five functions in machine code. Function 1 provides the optical properties of the new glass pane. Function 2 provides the optical properties of the whole BIST element. Function 3 is the energy simulation model of the BIST element. Function 4 is the energy simulation model of the building including the BIST element and the building services. Function 5 is a tool for the facility management, which indicates whether the BIST system performs as expected.*

If machine-code functions are to be exchanged, there are various formats in which they can be exchanged. They could be within an executable module with an input and an output file, but in this case, every simulation iteration would open a new process resulting in increased overhead cost, which may have a noticeable impact on parts that are critical to performance. The machine-code functions can also be dynamically linked, and thus distributed in a shared libray as a dynamic link library (DLL) or shared object (.so). In that way, the overhead cost is reduced compared to an .exe. Moreover, dynamic linking is a practical way of using libraries under GNU Lesser General Public License. It is essential that an entry point of the function and its arguments and their formats is provided along with the machine-code function. It is also recommended to provide documentation which explains the function and the arguments for the users in order to avoid wrong usage of the function. The function may be used in various simulation environments and in combination with several programming languages.
It could therefore be agreed to not only share the minimum requirements (the function and its description), but also an example of how to use it in a specific programming language or simulation environment in order to make it easier to use the functions. The exchange of a TRNSYS Type with a DLL as mentioned above, supplemented with a proforma file and an example deck, is one way of exchanging machine-code functions which are easy to use in a single simulation environment. For this approach, it is recommended to use a single source code with options to generate machine code for different environments, thereby transforming an existing simulation model into a multi-environment simulation model.
If the properties and equations of the product are not proprietary knowledge, the simulation pattern could be shared as open code within IFC. While the current capabilities of the IFC format do not provide a robust way to link input data with easily sharable functions. For example : we can imagine, in addition to the IFC file or preferably within the IFC file, a EXPRESS scheme file [28] that would carry the product entities with input as attributes, output as derived attributes, and defined functions that link them. If simulation environments start supporting EXPRESS scheme capabilities, we will be able to automate the creation of simulation models (like the TRNSYS files described earlier) from an IFC file. Furthermore, we would be able to share the simulation model by exporting it back into IFC, so other simulation environments would be able to use it. A similar solution can be implemented using the IFC Procedural language to carry an IfcProgram that defines the functions in open code [29].
Another approach, using already existing solutions in the simulation world, is the standardization of exchanged machine-code function and the description in text format combined in a Functional Mock-up Units (FMU) as proposed by the Functional Mock-up Interface standard [30]. The standardization allows various programs to read the description and use the function automatically. The description of the function can include all the details of the function, such as all equations. However, this should not be mandatory in order to allow confidentiality of trade secrets within the function.

For each building process, one method of data exchange would be desirable. There are countless desirable ways of exchanging information for different building processes. Therefore, the version of the Industry Foundation Classes should ideally allow all proposed methods of exchange.
With the current IFC version 4, solutions and workarounds are already possible, such as saving the machine-code function separately from the IFC file and linking it via IfcExternalReference. The problem here is that one always has to make sure that the path remains consistent. Small machine-code functions could be saved in text format in IFC4 as IfcText but this is limited to 32767 octets [31]. A better solution would be to make use of the new data type IfcBinary, introduced in IFC4Add1. Since IFC4Add2, its usage is possible in the context of a property as a type for an IfcSimpleValue. This enables a close connection of the properties of IFC components and the machine-code functions that characterise their properties.
From “big BIM” with all information on one server, as in [32], and “little BIM”, with conflict management between models on the individual computers of different stakeholders, down to very simple forms of data exchange such as email attachments, there are many possible physical locations for the machine-code function and exchange of the input and output data. One approach applies multi-agent systems [33], where programmed agents may run the machine-code functions automatically. Another approach are processes which are triggered under certain conditions e.g. when committing changes to the BIM trunk. Depending on the implementation and structure, BIM databases can evolve from one BIM available in one server into a network BIM, where fragments of the database can be stored in different physical locations to ensure that confidentiality and financial interests are not compromised. For example: A building envelope manufacturer and an HVAC system manufacturer can save the simulation functions of their products in their respective servers. They will not have access to the complete building information but only to the minimum amount of data needed for cost assessment and feasibility studies. Such data would be usually be in the general contractor’s server. Simulation professionals would be able to simulate the performance of the manufacturer’s solutions by getting the building model from the general contractor and recreating it in a simulation environment. To simulate the performance of the building, they would still need to know the behaviour function of the envelope and the HVAC system. In order to get it, the simulation environment would calculate the input of the functions and send them to the manufacturer’s servers, which would use the simulation function and send back the output. Such a workflow, although it may be slower and need greater investment, can lead to a very secure simulation environment that communicates with multiple servers e.g. by using processes which are triggered under certain conditions. The speed of such simulations can be drastically increased in the case of simulations with independent steps. Daylighting simulations, for example, are often independent for each time step, and therefore can send the input of multiple steps at once to the server. The communication delay between the client and server would not have a noticeable impact on the simulation time. To implement this solution, we can rely on IfcDocumentReference. Its “Location” attribute can hold the machine code’s location within the manufacturer’s server. Its “ReferencedDocument” attribute holds data about the referenced document, and it has itself a “Confidentiality” attribute which can be used to ensure that every project partner has access only to the necessary references.

## 4. Discussion

As a first step, it was proposed to introduce properties into the next IFC version that may be needed in cost-effective building processes of high-quality buildings. Some effort is necessary to determine which properties should be included and how to define them. Then it should be rather easy to introduce them as a new property set for high-quality buildings. Conventional building processes without solar systems would not be affected, because this new property set can be ignored when it is not needed.
The result section then proposed in general to include machine-code functions in the data exchange of building processes. If this is done intelligently, it could make high-quality buildings more cost-effective because

- more knowledge is shared due to the confidentiality of the machine code
- less time is needed for variations since no export to text and import from text format is needed
- the greater accuracy of detailed models can be used instead of simplified models
- more competition between products from different sources due to the modularity of the functions
- fewer mistakes while exporting and importing models by using the same validated model

In general, the transition from building processes without exchange of machine-code functions needs the effort of adaptation. If the machine-code functions are exchanged with poor quality, a lot of effort may be needed to use functions of other stakeholders, large computing times may result and the functions may generate incorrect results. However, these disadvantages can already appear in high-quality buildings today, where the building processes need to adapt to the innovative components, where much effort may be needed to include these components, where long computing times may result and mistakes may cause incorrect results.
The result section then proposed several ways of exchanging machine-code functions. In general, the next IFC version should offer many ways of exchanging code and machine code because of the reasons mentioned above and because it is difficult to exchange information within the building process if IFC is not applied. Such an

extension would not affect conventional building processes because it can be ignored, but it could make the building process of high-quality buildings more cost-effective when it is executed well.

The minimum exchange of machine code involves the function and a description of the name and the arguments of the function. It has the advantage that it is easy to write and to document and the drawback that some effort can be necessary to use the function. Adding an example to illustrate usage of the function needs little effort if it is only for one simulation environment or programming language and is very helpful for using the function. Examples for many simulation environments and programming languages first need the effort of coupling a function to this environment or language. After that, it is rather easy to provide many examples.

Sharing functions as an executable module may be helpful for certain cases where the inputs and the outputs will be checked and modified by a human being. However, they cannot be recommended if computing times are crucial and the function is called often. Sharing machine-code functions as a shared object (.so) or DLL has the drawback that much random-access memory (RAM) may be needed if only a small part is needed of many large functions. Using adaptive simulation models with variable accuracy and smart usage of the functions, the shared object/DLL approach offers short computing times even for nested functions when the RAM is not the limiting factor.

Documentation with detailed explanation of the variables used by the function and how the function works may need substantial effort but it reduces the risk of incorrect usage of the function. It is therefore recommended in cases where the arguments of the function are not obvious but represent complex quantities that need to be understood and used correctly.

A standardised method of providing the minimum exchange of machine code can make the implementation in several simulation environments much easier. It needs to offer the option of confidentiality and it would be much cheaper than generating implementations for many simulation environments. Just one implementation of the standardised interface would be needed in each simulation environment. However, some innovative components are difficult to couple with the existing models, especially if parts of the source code of the simulation environment are closed. Therefore, the option of sharing implementations of one function for specific environments is necessary. The effort for implementation in a new environment or version of the same environment may be substantial, but as long as an implementation of a function works well, it is easy to provide implementations of similar functions.

It is also possible to share the entire source code when no business secrets are divulged. In some cases, it is very easy to include a certain source code in a certain simulation environment. In other cases, compiling the source code and linking it to the rest of the simulation may need similar effort to receiving a machine-code function and implementing this function.

The confidentiality of a machine-code function depends on the effort needed to decompile the function and disclose the business secrets. The detailed physical model of [19] uses about 1400 variables. Without knowing the names of the variables and the nested functions, it will be difficult to determine the physical details of the transparent solar thermal collector by decompiling the DLL. Simple BIST models as presented in [34] can be built with a few lines of code but they typically include empirical parameters which do not provide much information about the technical details of the component. In any case, manual and automated obfuscation can be used so that the effort of reverse engineering increases substantially. If even this level should not be enough due to the importance of the business secret and the computing time is crucial, it is recommended to develop a simpler model from which the technical details could not be extracted even if the full source code were available. Another option is to store the machine code on a safe server and to exchange only the inputs and outputs with the users.

If the machine-code functions can run automatically when certain conditions are true, this could reduce the costs of the building process further. In many cases, human beings need to modify the existing simulations. However, if updates are needed, often automation is well suited for this.

## 5. Conclusions

This paper first analysed existing experience with the use of machine code functions in a building process including a new building-integrated solar thermal façade element. For the next version of the Industry Foundation Classes, a new property set for high-quality buildings and the inclusion of machine-code functions were then proposed. For the machine-code functions, several formats were proposed and their inclusion in the next IFC version was recommended. Finally, the advantages and disadvantages of the proposed changes were discussed. As several formats have specific advantages and disadvantages, the next IFC version should not exclude any of them but offer them all as options. The paper has focused on building-integrated solar systems, which are more complex than conventional building envelopes. However, machine-code functions offer a method to make their building process more cost-effective and reliable also for other complex building components. The exchange of machine-code functions may contribute also to other high-quality buildings using innovative components, for example, with adaptable building envelopes without active solar systems. For simple conventional buildings, machine-code functions may not be needed.

## 6. Acknowledgements

The authors thank Helen Rose Wilson for proof-reading the manuscript. This work was funded by the German Federal Ministry for Economic Affairs and Energy (BMWi), based on a decision by the German Bundestag.

## 7. References

[1] H.-M. Henning, A. Palzer, A comprehensive model for the German electricity and heat sector in a future energy system with a dominant contribution from renewable energy technologies – part I: methodology, Renew. Sust. Energ. Rev. 30 (2014) 1003–1018.
[2] A. Palzer, H.M. Henning, A comprehensive model for the German electricity and heat sector in a future energy system with a dominant contribution from renewable energy technologies – Part II: Results, Renew. Sust. Energ. Rev. 30 (2014) 1019–1034.
[3] European Parliament, European Council, Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings (recast), Official Journal of the European Union 53 (L153) (2010) 13–35.
[4] EnEV, Verordnung über energiesparenden Wärmeschutz und energiesparende Anlagentechnik bei Gebäuden (Energieeinsparverordnung - EnEV): Bundesgesetzblatt Teil 1; Nr. 59, Seiten 3029 - 3104, 2001.
[5] EnEV, Zweite Verordnung zur Änderung der Energieeinsparverordnung: Bundesgesetzblatt Teil 1; Nr. 67, Seiten 3951 - 3990, 2013.
[6] EnEV, Verordnung zur Änderung der Energieeinsparverordnung: Bundesgesetzblatt Teil 1; Nr. 23, Seiten 954 - 989, 2009.
[7] EnEV, Verordnung über energiesparenden Wärmeschutz und energiesparende Anlagentechnik bei Gebäuden (Energieeinsparverordnung - EnEV): Bundesgesetzblatt Teil 1; Nr. 34, Seiten 1519 - 1563, 2007.
[8] EnEV, Bekanntmachung der Neufassung der Energieeinsparverordnung: Bundesgesetzblatt Teil 1; Nr. 64, Seiten 3146 - 3162, 2004.
[9] U.S. Green Building Council, Leadership in Energy and Environmental Design (LEED), available at http://www.usgbc.org/leed (accessed on August 23, 2016).
[10] Building Research Establishment, Building Research Establishment Environmental Assessment Method (BREEAM), available at http://www.breeam.com/ (accessed on August 23, 2016).
[11] Deutsche Gesellschaft für Nachhaltiges Bauen, German Sustainable Building Council (DGNB) certificates, available at http://www.dgnb.de/dgnb-ev/en/ (accessed on August 23, 2016).
[12] C. Maurer, C. Cappel, T.E. Kuhn, Progress in building-integrated solar thermal systems, in: Elsevier (Ed.), Progress in Solar Energy, submitted.
[13] buildingSMART, Homepage of the international non-profit organization buildingSMART, available at http://buildingsmart.org/ (accessed on July 26, 2016).
[14] ISO 16739:2013, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries(ISO 16739:2013), International Organization for Standardization, available at http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51622 (accessed on July 26, 2016).
[15] ISO 12006-2:2015, Building construction - Organization of information about construction works - Part 2: Framework for classification (ISO 12006-2:2015), International Organization for Standardization, available at http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=61753.
[16] ISO 12006-3:2007, Building construction - Organization of information about construction works - Part 3: Framework for object-oriented information (ISO 12006-3:2007), International Organization for Standardization, available at http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38706.
[17] T.E. Kuhn, Cost-Effective: Converting façade of existing high-rise buildings into multifunctional, energy gaining components. Homepage, available at http://www.cost-effective-renewables.eu/ (accessed on August 23, 2016).
[18] C. Maurer, T.E. Kuhn, Variable g value of transparent façade collectors, Energy and Buildings 51 (2012) 177–184.
[19] C. Maurer, Theoretical and experimental analysis and optimization of semi-transparent solar thermal façade collectors. Dissertation, Freiburg, 2012.

[20] C. Maurer, T. Baumann, M. Hermann, P. Di Lauro, S. Pavan, L. Michel, T.E. Kuhn, Heating and cooling in high-rise buildings using facade-integrated transparent solar thermal collector systems, Journal of Building Performance Simulation (2013) 1–9.
[21] W.A. Beckman, L. Broman, A. Fiksel, S.A. Klein, E. Lindberg, M. Schuler, J. Thornton, TRNSYS The most complete solar energy system modelling and simulation software, Renew. Energ. 5 (1-4) (1994) 486–488.
[22] MathWorks, MATLAB (MAtrix LABoratory) numerical computing environment, available at http://www.mathworks.com/products/matlab/ (accessed on August 23, 2016).
[23] C. Maurer, T. Pflug, P. Di Lauro, J. Hafner, F. Knez, S. Jordan, M. Hermann, T. Kuhn, Solar heating and cooling with transparent façade collectors in a demonstration building, Energy Procedia 30 (2012) 1035–1041.
[24] C. Maurer, D. Gasnier, T. Pflug, P. Plešec, J. Hafner, S. Jordan, T.E. Kuhn, First Measurement Results of a Pilot Building with Transparent Façade Collectors, Energy Procedia 48 (2014) 1385–1392.
[25] 5D-initiative, Process map (accessed on August 24, 2016).
[26] S. Jordan, J. Hafner, T. Kuhn, A. Legat, M. Zbašnik-Senegačnik, Evaluation of Various Retrofitting Concepts of Building Envelope for Offices Equipped with Large Radiant Ceiling Panels by Dynamic Simulations, Sustainability 7 (10) (2015) 13169–13191.
[27] S. Jordan, J. Hafner, T.E. Kuhn, A. Legat, M. Zbašnik-Senegačnik, Indoor environment in retrofitted offices equipped with radiant ceiling panels, GRAĐEVINAR (Journal of the Croatian Association of Civil Engineers) 68 (02) (2016) 125–134.
[28] ISO 10303-11, Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual (ISO 10303-11:2004), International Organization for Standardization, 2004, available at http://www.iso.org/iso/catalogue_detail.htm?csnumber=38047 (accessed on July 19, 2016).
[29] J. Amann, A. Borrmann, Embedding Procedural Knowledge into Building Information Models: The IFC Procedural Language and Its Application for Flexible Transition Curve Representation, Journal of Computation in Civil Engineering (2016).
[30] fmi-standard.org, Functional mock-up interface for model exchange and co-simulation, 2014.
[31] buildingSMART, Industry Foundation Classes IFC4: Release 4, available at http://www.buildingsmart-tech.org/ifc/IFC4/final/html/.
[32] C. Leifgen, T.E. Kuhn, U. Rüppel, J. Teizer, BIM-based Collaboration Platform for the Holistic Integration of Energy Active Façade Components, Proceedings of ICCCBE (2016).
[33] A. Wagner, U. Rüppel, Cooperative Design Methods for Energy Active Façade Elements Based on BIMserver and Multi-Agent Services using JADE, Proceedings of EGICE (2016).
[34] C. Maurer, C. Cappel, T.E. Kuhn, Simple models for building-integrated solar thermal systems, Energy and Buildings 103 (2015) 118–123.