

---

# BIM-BASED GENERATION OF MULTI-MODEL VIEWS

---

Peter Katranuschkov, Senior Researcher, Peter.Katranuschkov@tu-dresden.de

*Institute of Construction Informatics, Dresden University of Technology, Germany*

Matthias Weise, Senior Developer, Matthias.Weise@semproc.de

*Semproc, Dresden, Germany*

Ronny Windisch, Research Assistant, Ronny.Windisch@tu-dresden.de

*Institute of Construction Informatics, Dresden University of Technology, Germany*

Sebastian Fuchs, Research Assistant, Sebastian.Fuchs@tu-dresden.de

*Institute of Construction Informatics, Dresden University of Technology & TragWerk Software, Dresden, Germany*

Raimar J. Scherer, Professor and Institute Head, Raimar.Scherer@tu-dresden.de

*Institute of Construction Informatics, Dresden University of Technology, Germany*

## ABSTRACT

Building information modeling (BIM) introduces a new work paradigm that enables improved interoperability and better coordination of AEC processes. The current IFC standard provides a core reference model for that purpose along with a growing number of supporting domain extensions. However, generation of discipline and task specific model views from a general-purpose BIM based on IFC is still a difficult problem. Open challenges include the achievement of flexible tool-supported specification of domain model views but also the generation of ad-hoc multi-model views combining BIM data with data from other (non BIM) models such as costs, equipment etc.

In this paper we describe an overall approach for the generation of BIM-based multi-model views that meets such objectives. The realized reference implementation builds upon the use of an open IFC toolset developed at the University of Weimar and HOCHTIEF and a formal model subset definition schema (GMSD) and supporting tools initially developed by the first two authors. We extend the GMSD definitions on instance level by applying semantic queries meant to be resolved at runtime. Moreover, using XML schema based representations we provide for simultaneous import of BIM and non BIM data that can be queried, selected and filtered via harmonized cross-model operations.

**Keywords:** BIM, IFC, model filtering, multi-model views, BuildingSMART

## 1 INTRODUCTION

Building information modeling (BIM) introduces a new work paradigm that enables improved interoperability and better coordination of AEC processes. The current IFC standard (IFC2x4) provides a core reference model for that purpose along with a growing number of supporting domain extensions and envisaged model views (subschemas). Such views are necessary because IFC-based BIM is a large model schema comprising hundreds of object types and thousands of properties which makes it hard to use and difficult to validate, and because most applications have only a limited area of interest with regard to the full model. However, generation of discipline and task specific model views from a general-purpose IFC-based BIM is still a difficult problem. Efforts related to the development of an Information Delivery Manual (IDM, Wix 2006) and Model View Definitions (MVD, Hietanen 2006) suggest solutions but reveal also gaps that need to be overcome. Open challenges include: (1) formal and more flexible tool-supported specification of *domain model views*, (2) mechanisms for generation of *ad-hoc views* on object instance level, and (3) *multi-model views*, combining BIM data with data from other (non BIM) models, such as costs, equipment, supply chain etc. nD modeling efforts extending BIM with various other project-related data only partially give answers to these three issues.

The first issue arises from the need to speed up development and standardization of BIM submodels to provide for broader coverage of user requirements and information flows in the full lifecycle of AEC projects. The second issue addresses the use of BIM (and non BIM) data in specialized applications developed mainly by SME vendors that typically need only small portions of the full models. The third issue addresses similar problems but in a broader scope, trying to overcome the limiting requirement of a common all-encompassing model and respectively shared project data repository, a requirement that cannot be easily arranged for in most practical construction contracts.

In this paper we describe a new approach for the generation of BIM-based multi-model views that meets such objectives. Reported is work that started with an initial prototype development back in 2003 and is now extended in the lead German project Mefisto (Scherer 2009) and the EU project HESMOS (HESMOS 2010).

## 2 METHODOLOGY

A methodology for multi-model view development, implementation and use has to consider the overall process and the principal actors involved in it. Moreover, the lifecycle of the constructed facilities has to be taken into account, i.e. how and for what purposes are model views needed.

Generally, there are three groups of actors taking part in the process: (1) modelers, (2) software developers, and (3) end users. The last group does itself contain many subgroups and actor types which is one of the main reasons to advance model view development and standardization.

IDM-MVD suggests a structured approach to the development of BIM views that brings together modelers, end users and software developers in the frames of the BuildingSMART initiative. The principal process is illustrated in Fig. 1a adapted from (Grobler 2010). However, this figure also hides two substantial issues that need to be considered. Firstly, the IDM-MVD approach only provides for manually developed specifications. The process is hardly supported by any specialized tools and no automated validation of the developed subschema definitions is foreseen. Secondly, it is assumed that once implemented, model views can be readily used in practical projects. However, there is no specific support by open tools and methods that would enable applications to achieve the needed functionality. Hence, even though the process is clear, existing gaps can seriously limit its practical applicability.

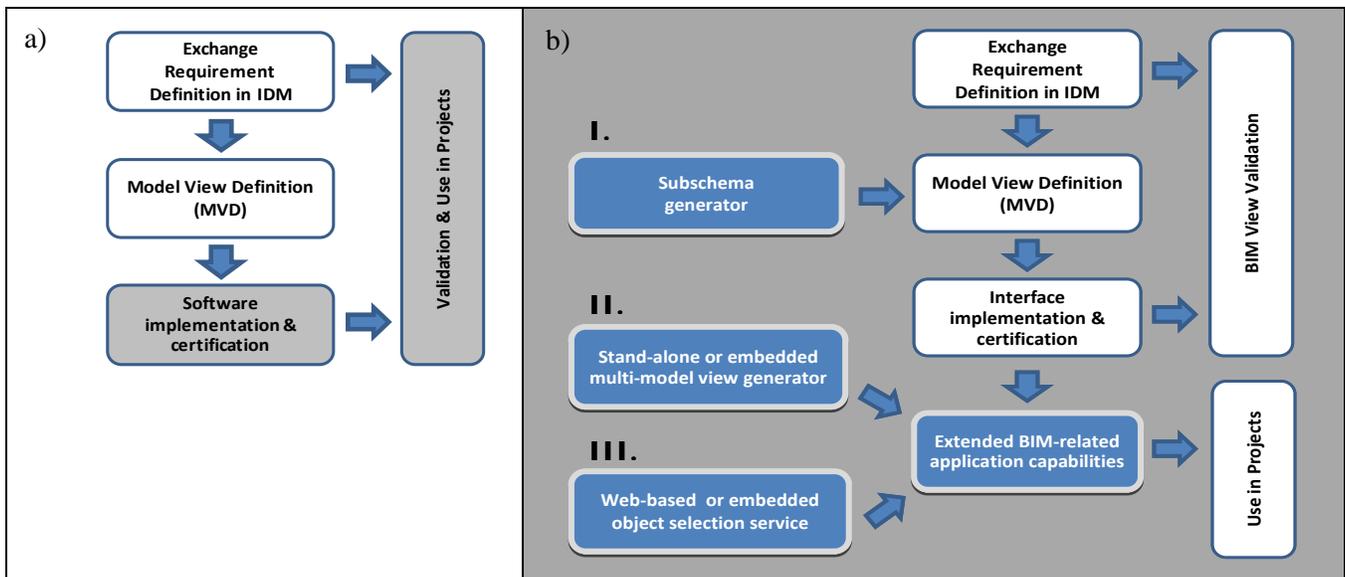


Fig. 1: High-level schema of the overall process of development, implementation and use of model views  
 a) According to BuildingSMART (Grobler 2010); b) Suggested extended methodology and tools.

To tackle these problems, we suggest extending the IDM-MVD methodology by *three additional steps*, as shown schematically on Fig. 1b. The first step provides for formal specification of a subschema, defined and validated with the help of software tools, which can be immediately used as basis for the development of interfaces for contracted data exchange. This would facilitate specifically the work of modelers and can strongly reduce model

view development time. The second and third step provide for fine-tuning a model view in the actual context of an application, enable interoperable consideration of non BIM data and facilitate end user operations on the model. These steps are directed to software developers but at the same time lead to better and more efficient user support. Together, these extensions allow also to clearly separate validation and use of model views which is not so evident in the pure IDM-MVD approach (compare Fig. 1a and 1b).

However, realization of the suggested methodology requires new methods and tools for all three outlined extensions. Indeed, while there are some efforts related to each of these extensions, there is no known approach yet attempting a complete solution. Regarding the generation of subschemas the work of Yang and Eastman (2007) applying a rule-based method can be mentioned. For the second and third proposed extensions most relevant is the work of Adachi who developed a formal query language for IFC-based BIM named PMQL (Adachi 2002). In the area of multi-model views most efforts have been focused on extending BIM to integrate the needed non BIM data, e.g. for scheduling (Tulke et al. 2008). However, this may increase and not reduce complexity.

In contrast to these efforts, we are developing a consistent *reference implementation for all suggested extensions* that can serve as proof of concept and at the same time offer a first open software solution with practical applicability. It is based on a formal model subset definition schema (GMSD) and a set of software tools providing the implementation methods (ViewEdit, Multi-Model View Generator, IFC Viewer and Open IFC Tools). The first of these tools was initially developed by the first two authors and is currently being extended with new functionality. The second tool is a new development being realized as part of a broader multi-model management framework (Fuchs et al. 2010). IFC Viewer and the Open IFC Tools ([www.openifctools.org](http://www.openifctools.org)) are developments of the University of Weimar and HOCHTIEF (Theiler et al. 2009), integrated into our multi-model framework. However, the core of the methodology and its most general and reusable part is represented by GMSD (Weise et al. 2003). GMSD is not a language but a schema that provides for a neutral definition format with possible mappings for various practical data exchange implementations. It is comprised of two separate, yet inter-related parts: (1) *view definition*, and (2) *object selection*. The first supports primarily the first suggested extension from Fig. 1b, whereas the second is related to the second and third extension for a single EXPRESS-based model, i.e. (but not limited to) IFC. Both parts can be used together, but view definitions would normally be used by modelers, whereas object selection is more appropriate for software developers – as embedded application feature or outsourced as external service.

Fig. 2 below presents the top level constructs of the GMSD schema indicating the envisaged scenarios of its use. How this is done in detail is explained in the following chapters.

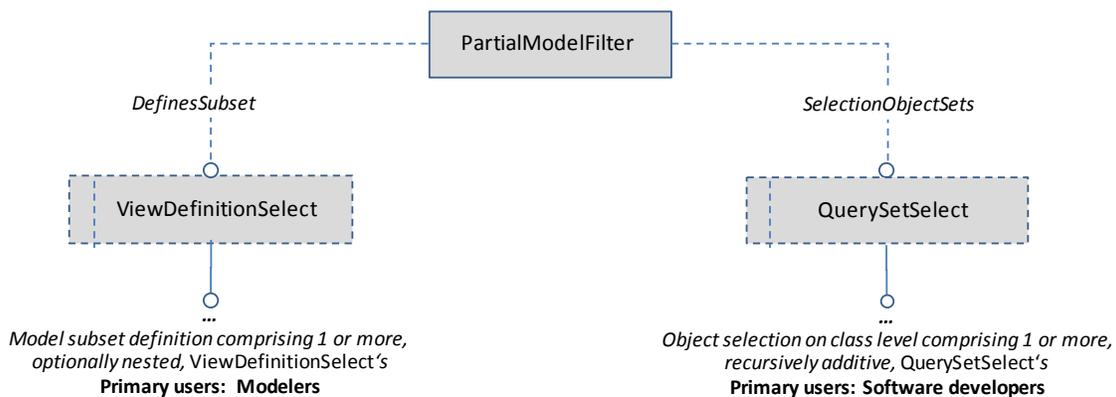


Fig. 2: The top-level constructs of the GMSD schema

### 3 CASE EXAMPLE

The outlined general methodology can be useful in many situations. To focus the discussion we present a practical example taken from the new European project HESMOS. It refers to the energy management of an existing office building during the operation phase (see Fig. 3 below). In the paper we examine specifically the control of the air conditioning system via temperature and humidity sensors for the case of rooms with sensible IT devices (server rooms). This requires the use of at least two data models: (1) the model of the building automation system (BAS),

and (2) BIM. Additional model data in that context are Costs, to enable “costs - energy efficiency” analyses. Using this example, in the next two chapters we discuss the suggested methodology in greater detail. However, various further examples from HESMOS and from the Mefisto project have shown that many different use cases and business processes can be supported by the same approach.

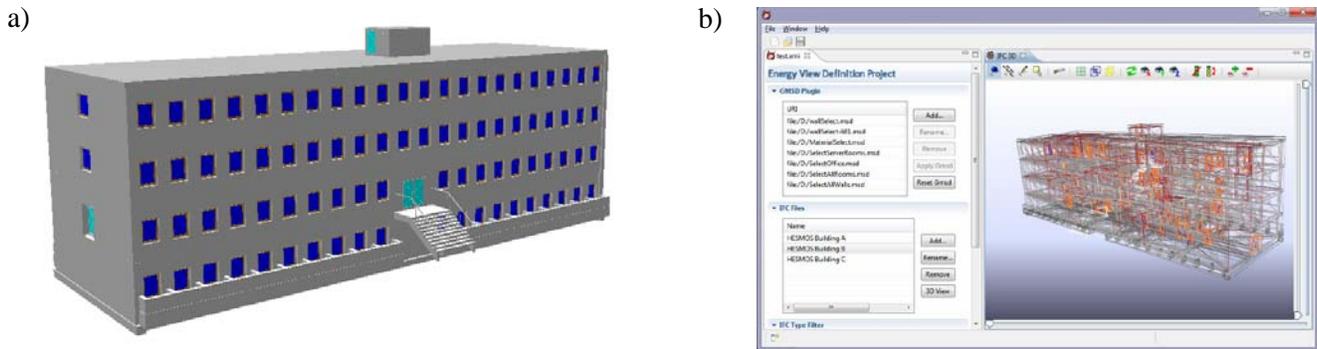


Fig. 3: Multi-model view example – BIM-based energy efficient building management  
 a) 3D view of the example building, b) the same building as wireframe model shown in the GUI of the multi-model view generator (currently under development at the TU Dresden)

## 4 BIM-RELATED VIEW GENERATION

BIM is by far the most complex model that has to be dealt with in a construction project. The current IFC2x4 model comprises 800+ object classes as compared to known specification or cost models, such as the German GAEB, featuring less than 50 object classes. Therefore, BIM-related view generation plays a central role in the overall interoperability problems that need to be solved. Essentially, this requires methods and tools for stepwise definition of partial model views by means of appropriate data filtering on schema, class and instance level.

### 4.1 Filtering on Schema Level

The main purpose of schema level filtering is the generation of valid subschemas of the large BIM schema, which can be used as basis for certain predefined data transitions such as model handover from architecture to structural or building services engineering, from design to construction, from construction to facilities management and so on. Whilst IDM is an obvious first step in that process and MVD can be well used to document and hence standardize the developed subschema(s), without a formal approach to subschema generation validation would remain a long-lasting, complex and uncertain task. This can be overcome with the help of a language or data model that explicates subschema definition, and a tool built on that basis which takes the full schema as input and provides a valid subschema as output. In terms of GMSD this is done by using one or more *ViewDefinition* specifications. In our reference implementation such specifications can be interactively developed within the Java-based *ViewEdit* tool which provides the option of creating the subschema corresponding to the developed GMSD view specification.

Fig. 4 shows the EXPRESS-G diagram of the part of the GMSD schema dedicated to the definition of static model views (subschemas). A *view definition* is always specified on class level. In GMSD it can be used to:

- Sort out object instances by their type or reduce their content on attribute / feature level
- Sever references to unneeded objects to create a more meaningful subset for the targeted domain
- Generalize objects to supertypes to lower the complexity of the resultant model subset
- Control or limit object references in accordance with some specified criteria.

The top level class *ViewDefinition* provides a unique name to identify predefined views, a Boolean flag indicating if referenced entities should be included by reference or by value, the names of the schemas for which the view definition is valid, a predefined (default) entity usage and extensions to override the defaults. With the *PredefinedEntityTypeEnum* the handling of object types can be set to (1) use all object types with all their features, (2) use all object types with only those features that are needed to keep data integrity, or (3) omit all object types and rely only on the extension specifications. Detailed definitions can then be provided by a set of

*EntityViewDefinitions* enabling the handling of each object type on feature (attribute) level. *FeatureSelections* can be specified to fine-tune the usage of each feature, whereby the available feature usage types (use, conditional use, delete) can be applied multiple times. Such alternative definitions can be assigned to referenced objects to allow for different interpretations, depending on the actual reference paths. This can be useful in cases where the reference path is important to process the actual object instances properly. For example, in the IFC model the processing of geometry provided via *IfcRepresentation* can be given in dependence of the element type (*IfcBeam*, *IfcWall* etc.) using the particular referenced geometry objects. Additionally, through the attribute *GeneralizeTo*, an object type can be typecast to one of its supertypes to enable more general representation.

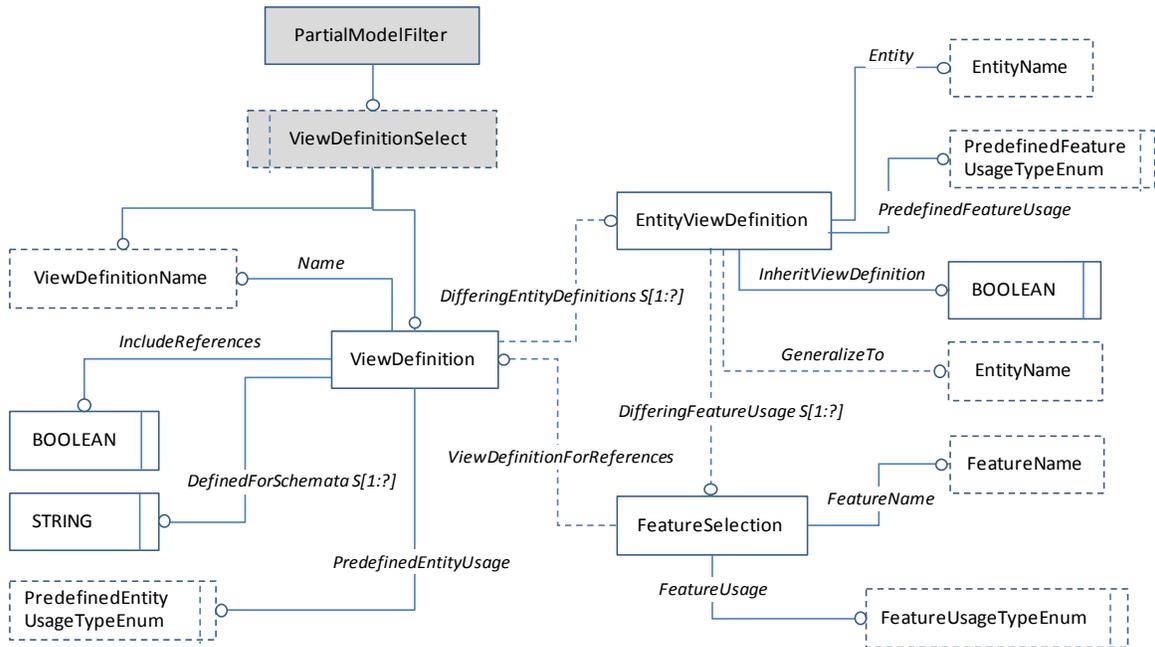


Fig. 4: Partial GMSD Diagram related to View Definitions (elements of minor importance are omitted)

Writing such a GMSD specification manually is fully legitimate, but it is not an easy and pleasant task. Therefore the *ViewEdit* tool was developed, greatly facilitating the process (Fig. 5, Scherer et al. 2006).

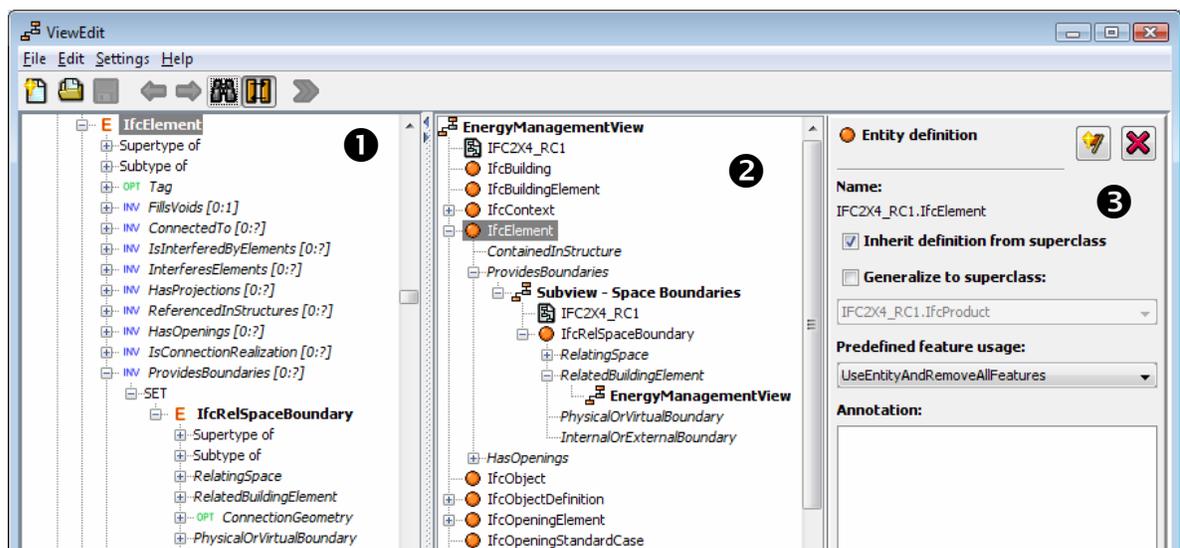


Fig. 5: Partial screenshot of ViewEdit showing the definition of the Energy Management subschema for the discussed case example

As Fig. 5 shows, the ViewEdit screen is divided in three panels: (1) a view of the source BIM with all its entity types (classes) and attributes (features), (2) a tree view of the developed view definition on class level, including subviews for the handling of referenced objects, and (3) an entity definition panel allowing to set various attributes of the item highlighted in panel 2. Specifically, the developed Energy Management subschema omits object classes that are not relevant for the targeted energy management tasks, such as building foundations and staircases as well as various non geometric attributes of other classes. Fig. 6 shows screenshots of the initial model and the obtained result, taken from the multi-model view generator integrating the IFC Viewer developed at the University of Weimar.

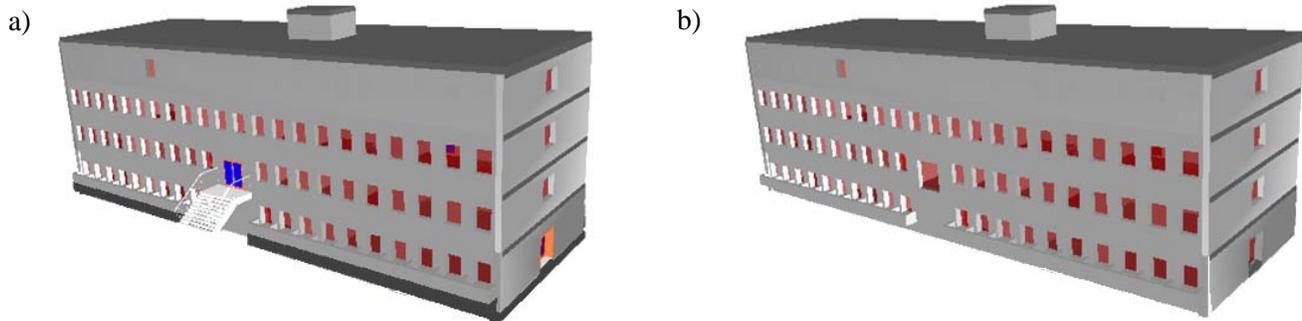


Fig. 6: Schema level filter for the given case example  
 a) Before applying the filter, b) After filtering: removed all stairs, foundations, various (non visible) entity attributes

There are several usage scenarios where schema level filtering can be of benefit. The first refers to standardized subschema development in support of IDM-MVD (see Fig. 1). Currently, the ViewEdit tool is being extended for that purpose to automatically generate the EXPRESS subschema specified via GMSD. However, this requires additional checking and eventually extending the output schema with classes/attributes that are not defined by GMSD to ensure validity and consistency with the parent BIM schema. A second scenario envisages the development of *modular views* that can build upon each other, thereby enabling gradual refinement and coverage of a broader range of design/construction tasks. A third scenario foresees the use of schema level filtering as an external Web Service or directly within an application, together with the further filtering steps explained in the next sections. These three scenarios could also be combined to provide for even greater flexibility and interoperability.

## 4.2 Filtering on Class Level

Filtering on class level can provide for more focused results in the context of a specific task. Its main purpose is to complement schema level filters to further reduce the required data subset. It may be performed on the full BIM data but the main intention is to use a standardized subschema as input (e.g. a structural design subschema or the Energy Management subschema referenced above), which is then further refined as a new subschema option or used dynamically at run-time. With GMSD this is done by using one or more *QuerySetSelect* specifications based on a predefined object selection set. In our reference implementation such specifications are also done with the Java-based ViewEdit tool, together with or independent from schema level filtering.

Fig. 7 shows the EXPRESS-G diagram of the part of the GMSD schema dedicated to the definition of class level filters via *QuerySetSelects*. It offers two principal possibilities to select objects: (1) by the result of various *selection type constructs*, and (2) via the set operations *union*, *difference* and *intersection* applied to the result of another select operation.

Through the reference to *SelectTypeSelect* various selection constructs can be used, i.e.:

- Filtering the instances of a particular object class using specific attribute constraints (value ranges, comparison operations such as *thickness > 14cm*, or even a constraint function defined externally)
- Including all instances of a specific object class with or without the instances of its subclasses
- Including/excluding referenced object instances
- Presence/absence of certain references to other objects in the selection set.

Using *SetOperation* more global combinations are possible, such as building the set difference of two already available views, inverting/deselecting a view etc.

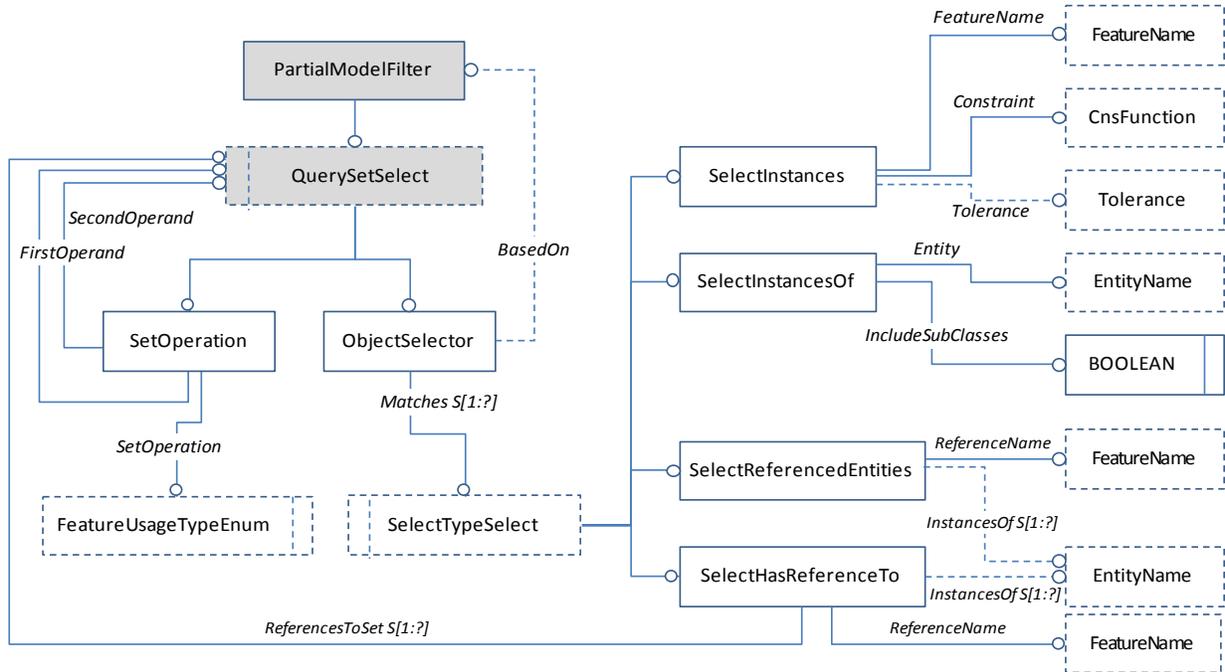


Fig. 7: Partial GMSD Diagram related to Class Level Filtering (elements of minor importance are omitted)

For the given case example, using a relatively simple GMSD filter the Energy Management subschema can be further simplified for the concrete targeted task, tracking the temperature and humidity in Server Rooms. In particular, the model is further filtered to contain only *IfcSpace* objects and associated object instances required for validity, and to the result selection set the attribute filter *Name = 'Server Room'* is applied. Fig. 8 below shows screenshots of the initial state (the result of the previous step) and the result of applying the class level filter.

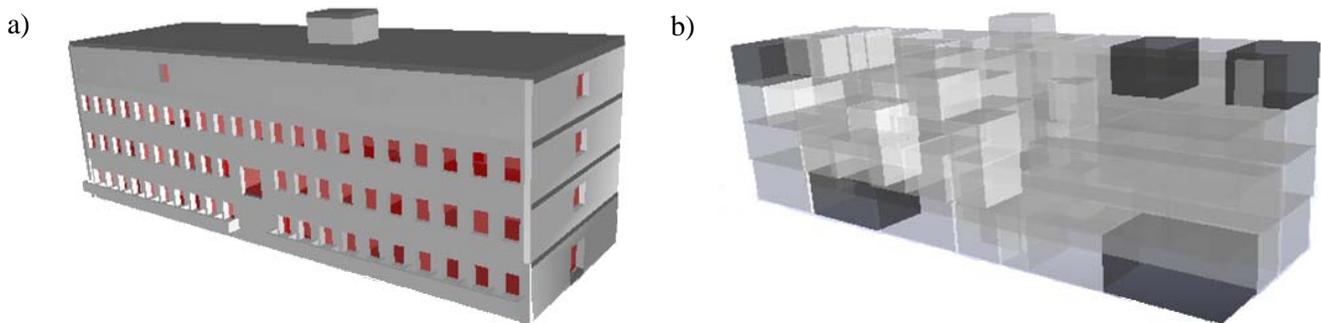


Fig. 8: Class level filter for the given case example  
*a) Before applying the filter, b) After filtering and highlighting the selected server rooms (IfcSpace instances); The resultant model is shown in semi-transparent mode to reveal all selected rooms in the building.*

The basic usage scenarios of class level filters are two. Firstly, they can be used for more focused specification of subschemas, providing an additional mechanism for development of subschema hierarchies. Secondly and more importantly, they can be used directly in applications, especially in cases where specialized solutions are the target, a typical market niche for SMEs in construction IT. This can be particularly useful if combined with capabilities for multi-model and object level filtering.

### 4.3 Ad-hoc Filtering on Object Level

Filtering on object level on the basis of specific object instance semantics cannot be realized by models like GMSD neither by tools like ViewEdit because they can only provide specifications on class/schema level. Such filters are

possible only in the particular context of specific application tasks. Their purpose is (1) to enable the users to select individual objects for further processing, and (2) to facilitate semantic single or multi-model filtering.

Principally, there are three ways of user-driven object selection:

- By supporting geometry constructs
- By topological queries
- By semantic queries.

*Geometric constructs* are well developed and widely known in CAD systems. However, they require implementation of sophisticated algorithms that are not in the focus of non CAD applications. Besides, they are mainly useful for applications that deal extensively with the geometry of BIM objects, which is not always the case, e.g. in costing, scheduling, construction management etc. In our reference implementation we use partially the geometry constructs developed in the extended version of the integrated IFC Viewer (Theiler et al. 2009). This includes cutting and clipping lines, planes and polygons as well as certain VRML-like view manipulation operations.

Use of *topological queries* can be very powerful provided that topology is supported in BIM. This can be done by applying a query language including constructs like “below”, “above”, “next to”, “inside” etc. A suggestion in that regard is done in (Borrmann and Rank, 2009). However, albeit foreseen in IFC, topological data are hardly supported by current BIM-CAD. Borrmann and Rank (2009) show how this can be overcome by additional reasoning but the suggested algorithms based on Octree geometry are complex and hence of limited applicability for non CAD tools. Thus, whilst definitely useful, topological queries are presently out of scope in our reference realization.

Last but not least, *semantic queries* can be applied to enable ad hoc focusing on the issues of interest. This requires: (1) an engineering query language that maps BIM constructs to the engineering concepts used in the targeted application domain, and (2) appropriate visualization techniques to highlight and zoom to the selected object(s) automatically. Regarding these issues, various suggestions have been worked out in the last years. Katranuschkov et al. (2003) have suggested a formal engineering language based on XML Schema, and developed a light-weight web-based prototype tool as proof of concept in the frames of the European ISTforCE project. A similar approach has been undertaken later by Nepal et al. (2009) in the area of construction management. The obtained results are promising but scalability of the XML definitions is yet to be examined. Nevertheless, with a general mapping schema between IFC and engineering concepts it is possible to realize dedicated queries in the scope of a specific application, thereby enabling goal-oriented object search and selection. This can be especially powerful for multi-model views where a single query can simultaneously address concepts and properties of multiple models.

## 5 MULTI-MODEL VIEW GENERATION

With methods and tools enabling filtering / reducing the complex building information model on three levels, multi-model views and their interoperability can be readily addressed. The purpose of such views is to take into account and integrate non BIM data provided on the basis of standard or quasi standard schemas without further expansion of the BIM. There are at least four reasons to avoid such expansions: (1) keep the models separately manageable, (2) avoid increase of the overall complexity, (3) enable use of state-of-the-art non BIM data such as GAEB XML, MS Project etc., and (4) enable straightforward data exchange with further downstream applications without additional mappings from the extended BIM back to the native schemas. However, in order to realize such a flexible approach, three additional issues have to be tackled – the filtering of non BIM models, the proper inter-linking of the separate single model views to provide for integrated interpretation of the data, and the provision of an adequate GUI for multi-model management.

In principle, the process of *filtering non BIM data* can follow the same general methodology described in the previous sections. Normally this would be faster and easier to achieve because non BIM data used in construction projects are of lower complexity compared to BIM. However, a practical difficulty provides the fact that almost no other model schema is based on EXPRESS, the basis of the IFC model. Therefore, while subschema development would be generally similar, the specific implementation methods must be adapted to the underlying representation.

*Combining multi-model views*, once such views are available, faces three problems: (1) tackling of the multiple views within one and the same application, (2) inter-linking these views to an interoperable system, and (3) storing these links for further reuse. The first is a pure technical issue on implementation level. The second requires finding the correspondences between the individual models which can be done automatically if there are

corresponding features allowing for the necessary mappings, or manually, if such features are absent. In both cases, the solution is strongly dependent on the actual models and has to be solved within the software using them. Most interesting for our subject is the third issue because it allows for a generic solution. For that purpose we suggest using a *Link Model* built upon XML Schema, which stores all correspondences between the model elements once these correspondences are found. Its essence is to couple pairwise instances of the related models using one or more key attributes as unambiguous references, whereby general model inter-relationships are defined on the top level of the XML schema. This lean model relies on the fact, that even if no link attributes are available in the original models the software can provide at least object IDs internally which can then be used to describe correspondences. However, such links must not be limited to IDs. Any attribute combinations can be used if they provide unambiguous inter-model references.

Last but not least, *querying the generated multi-model views* and providing an efficient user interface has to be addressed. Fig. 9 below, showing a screenshot of the multi-model view generator for the provided case example, can be used to explain the concept. It presents the last step of the task described by that example, namely finding the Server Rooms where temperature and humidity thresholds measured by room sensors are trespassed and require respective action. To do that, both BIM and BAS data are first mapped to engineering concepts, i.e. *IfcSpace* → *Room*, *Sensor.temp* → *T [Room]* etc., using the principal approach described in (Katranuschkov et al. 2003). These data can then be queried via pre-configured user interface forms, dynamically enabled via the multi-model framework being developed in the frames of the Mefisto project (Fuchs et al. 2010). Basis for the user interface are formal semantic queries that can address object instances and their properties *across* the individual model views.

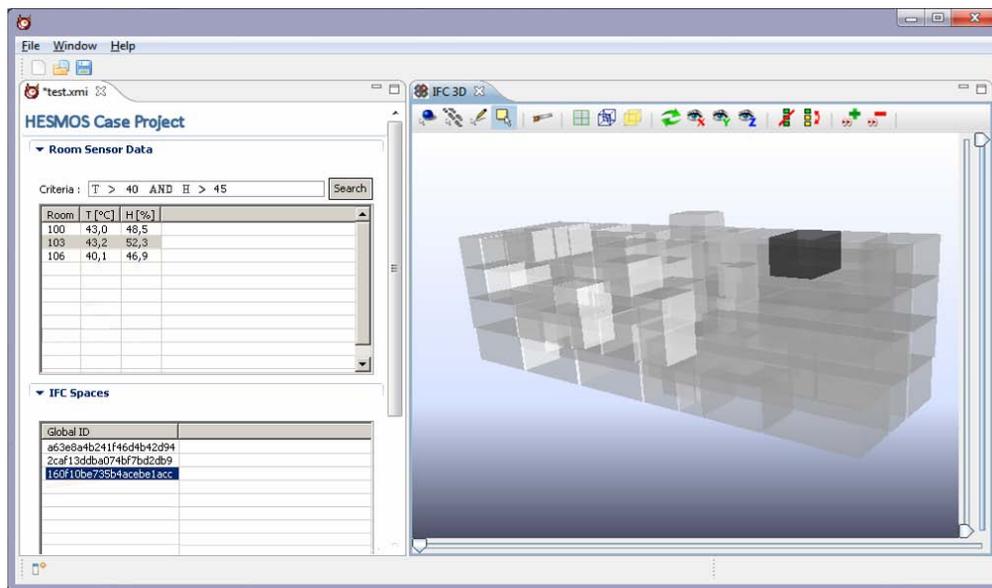


Fig. 9: The GUI of the multi-model view generator showing the last filtering step in the case example

The displayed query results in three (out of five) rooms that correspond to the given criteria (Fig. 9, upper left panel). Their IFC IDs are automatically determined (lower left panel) which enables highlighting the focused room (103) in the IFC Viewer (right panel) and examine relevant further features (not shown in this screenshot).

Thus, semantic and other multi-model queries complete the overall process enabling gradual and goal-oriented reduction of the data towards the desired area of interest. For the discussed example this process can be shortly expressed as follows: *Full BIM* => *IfcSpaces* => *Server Rooms* => *Server Rooms with T > 40 & H > 45*.

## 6 CONCLUSIONS AND OUTLOOK

We described a new methodology for multi-model view generation encompassing a structured overall approach as well as a set of enabling methods and tools for efficient and flexible view development and use. The presented methodology is coherent with the IDM-MVD efforts undertaken within BuildingSMART, extending these to better

support both modelers and software developers. Moreover, even though the objectives of the research are not explicitly addressing end user needs, the developed reference implementation already identified a number of scenarios verifying the approach and its benefits to practice. One such scenario was outlined in the paper but there are many more that can be envisaged, e.g. a costing application for subcontractors such as plumbers or roofers, BIM based construction management, BIM based energy simulation for architects, facility managers and public users and so on. However, whilst initial ideas and work on GMSD started back in 2003, there are still a number of issues that need to be considered. Planned or already under development in the frames of the Mefisto and the HESMOS projects are various GMSD extensions, especially with regard to class/attribute selection and constraints, elaboration of the filtering approach with regard to XML schema based models and the suggested link model concept, extension of the functionality of the ViewEdit tool, substantial work on the multi-model view generator which is yet in an early phase and so on. Upcoming results of this work will be reported in subsequent publications.

The presented research in this paper was enabled by the financial and technical support of the Mefisto project partners and the German Ministry of Education and Research as well as the HESMOS partners and the European Commission. Special support was provided by Prof. Beucke and Mr. Tauscher (Bauhaus University Weimar) and Dr. Tulke (Hochtief AG). This support is herewith gratefully acknowledged.

## REFERENCES

- Adachi, Y. (2002) "Overview of Partial Model Query Language", *VTT Report VTT-TEC-ADA-12*, Espoo, 12 p.
- Borrmann, A. and Rank, E. (2009) "Topological analysis of 3D building models using a spatial query language", *Advanced Engineering Informatics* 23(4), pp. 370-385.
- Fuchs, S., Katranuschkov, P. and Scherer, R. J. (2010) "A framework for multi-model collaboration and visualisation", *Proc. ECPPM 2010*, Cork, Rep. of Ireland, 14-16 Sept. 2010, Taylor & Francis, pp. 115-120.
- Grobler, F. (2010) "IDM-MVD: How do they provide solutions to user requirements?", © BuildingSMART 2010, <http://www.buildingsmart.com/files/MVD:IDM%20expl.pdf>
- HESMOS (2010) EU Project No. 260088 HESMOS "ICT Platform for Holistic Energy Efficiency Simulation and Lifecycle Management Of Public Use Facilities" – Description of Work.
- Hietanen, J. (2006) "IFC Model View Definition Format", © IAI, April 2006. [http://www.iai-international.org/software/MVD\\_060424/IAI\\_IFCModelViewDefinitionFormat.pdf](http://www.iai-international.org/software/MVD_060424/IAI_IFCModelViewDefinitionFormat.pdf)
- Katranuschkov, P., Gehre, A. and Scherer, R. J. (2003) "An Ontology Framework to Access IFC Model Data", *ITcon Vol. 8, Special Issue "eWork and eBusiness"*, pp. 413-437; <http://www.itcon.org/2003/29>, 2003.
- Nepal, M. P., Zhang, J., Webster, A., Staub-French, S., Pottinger, R. and Lawrence, M. (2009) "Querying IFC-Based Building Information Models to Support Construction Management Functions", in: *Proc. of the 2009 Construction Research Congress "Building a Sustainable Future"*, ASCE Publ., pp. 506-515.
- Scherer, R. J. (2009) "Mefisto: an ICT platform for foresighted and partnering-based Construction (in German)", Presentation at the project kick-off workshop, Berlin, July 2009, Available at: [www.mefisto-bau.de](http://www.mefisto-bau.de)
- Scherer, R. J., Weise, M. and Katranuschkov, P. (2006) "Adaptable views supporting long transactions in concurrent engineering", *Proc. Joint International Conf. on Computing and Decision Making in Civil and Building Engineering*, Montreal, Canada, June 14-16 2006, ISBN 2-921145-58-8, pp. 3677-3686.
- Theiler, M., Tauscher, E., Tulke, J. and Riedel, T. (2009) "Visualizing IFC-Objects with the help of Java3D" (in German: Visualisierung von IFC-Objekten mittels Java3D), in: von Both, P. and Koch, V. (eds.): *Forum Bauinformatik 2009*, Universitätsverlag Karlsruhe, ISBN 978-3-86644-396-9, pp. 149-159.
- Tulke, J., Nour, M. and Beucke K. (2008) "A Dynamic Framework for Construction Scheduling based on BIM using IFC", *Proc. 17<sup>th</sup> Congress of IABSE*, Sept. 17-19, Chicago, USA.
- Weise, M., Katranuschkov, P. and Scherer, R. J. (2003) "Generalised Model Subset Definition Schema", in: Amor R. (ed.) *Construction IT: Bridging the Distance, Proc. CIB-W78 Workshop, Waiheke Island, NZ*, 16 p.
- Wix, J., editor. (2006) "Information Delivery Manual: Guide to Components and Development Methods", Available at: <http://idm.buildingsmart.no>
- Yang, D. and Estman, C. (2007) "A Rule-Based Subset Generation Method for Product Data Models", *Computer-Aided Civil and Infrastructure Engineering* 22(2), pp. 133-148.