

MANIPULATING IFC SUB-MODELS IN COLLABORATIVE TEAMWORK ENVIRONMENTS

M. M. Nour

Bauhaus-Universität Weimar, Informatik im Bauwesen, Germany

ABSTRACT: This paper addresses the problem of workflow management in collaborative teamwork environments, where multidisciplinary actors and software applications are involved. Design versions or variants may represent different development stages, partial designs or solution alternatives that need to be integrated together. The paper presents a novel approach to splitting and merging IFC sub-models (partial models) at different degrees of granularity away from the schema oriented approaches. It relies on an instance oriented approach (FIOPE) Flexible Instance Oriented Partial Exchange Environment, that enables end users' manipulation of the IFC model and consequently the contribution to the evolution of exchange patterns which can be formalised later to partial exchange schemata (e.g. EXPRESS-X ISO 10303 P-14). The process of comparing IFC/STEP models' tree structure is neither simple nor straightforward. Consequently, certain algorithms are developed to overcome the complexity of the IFC EXPRESS-ISO P-11 definition of the IFC model. A whole range of software tools for reading, visualizing, processing and writing IFC/STEP models have been developed to achieve the above interoperability aim.

KEYWORDS: IFC, collaborative team work, partial models, interoperability.

1 INTRODUCTION

1.1 Interoperability

Since the mid nineties, the AEC/FM (Architectural Engineering Construction / Facility Management) community has allocated a lot of research efforts towards solving its Interoperability problems. Interoperability in the building industry is defined as “An environment in which computer programmes can share and exchange data automatically (without translation or human intervention), regardless of the type of software or of where the data may be residing” (NCCTP 2007). Another relevant definition from the (Merriam-Webster, 2007) dictionary is “The ability of a system to use the parts or equipment of another system.”. The Oxford Dictionary defines the term as a noun derived from the adjective *interoperable* “Able to operate in conjunction”. It is also defined as “The ability of two or more systems or components to exchange information and to use the information that has been exchanged.” (IEEE, 1990). It could be concluded from the above definitions that the main focus is on (1) data sharing, (2) the ability to exchange usable information and (3) platform independence, i.e. away from proprietary developments.

1.2 Industry foundation classes

The International Alliance for Interoperability (IAI) introduced the Industry Foundation Classes (IFCs – ISO/PAS

16739, 2005) as a non-proprietary standard for information exchange among AEC/FM project participants. It provides definitions for syntax as well as semantics of construction elements. It facilitates the exchange of both geometric as well as alphanumeric data of construction elements and their inter-relations.

The IFC model itself is defined in EXPRESS (ISO 10303-P11, 1994) schemata. EXPRESS is a lexical object oriented modelling language. It possesses a unique ability of enforcing rules and constraints on its objects that is hard to map one to one to other modelling or programming languages. For a thorough introduction to the language, the reader can refer to (Schneck and Wilson, 1994).

The data instances of the IFC model are exchanged in the STEP (Standard for Exchange of Product Model Data - ISO 10303-P21, 2002) format. It is the means by which data defined by an EXPRESS (ISO-10303-P11) schema can be transferred from one application to another. It is a clear text encoding of the exchange structure that represents data according to a given EXPRESS schema. A STEP file consists of two main parts; first is the HEADER section and then the DATA section(s). The header section includes information about the application that produced the IFC model, a time stamp, schema version, file name, file description and so forth. On the other hand the DATA section(s) consists of an arbitrary number of IFC elements. Each element is a mapping of the attribute values and data types of the EXPRESS definition of the element in a super to sub-class order.

1.3 Collaborative multidisciplinary teamwork

In the majority of construction procurement systems, design work needs to be done in a multidisciplinary teamwork environment. The design process is by nature illusive and iterative inside the same discipline and between other AEC disciplines. The design process possesses many versions that represent various design development stages. It also includes many alternative design concepts and decision making processes. An obvious general example is the delivery of architectural designs to structural and HVAC engineers while still undergoing further design refinements and iterations. It is even worse when we consider fast-track procurement routes.

In the sequential workflow, it is straight forward and there are no big problems as long as a lossless information handover from one stage to another is guaranteed. However, in parallel working approaches, severe problems related to data acquisition and management in addition to multi and inter disciplinary collaboration arise. Often design team members from the same discipline use different software tools and work in parallel. For example, a building can be divided into three different sections among three different architects to design. Every architect can be using a different software tool, and they have to integrate their work at the end.

2 STATEMENT OF PROBLEM

2.1 The use of building information models

Building Information Models (BIMs) in general and IFC in particular are aimed at achieving interoperability between software tools that are used in the entire lifecycle of a construction project. It is envisaged that all tools will be able to work on a central pool of project data. Although, the majority of AEC software developers have IFC APIs that are capable of importing and exporting IFC/STEP files, it is still not possible to make full use of the IFC model and abandon the file based exchange scenario. This is attributed to the fact that an IFC model of a certain project is exchanged as a whole unit. In the meantime, the internal structures of different software applications do not support the whole range of information that is covered by the IFC specifications. This makes it nearly impossible to maintain a lossless data exchange across applications.

2.2 Practical implementation aspects

From a work flow management perspective, in order to achieve a lossless information exchange among AEC/FM project members, either the software developers should change their internal data structuring to eliminate irrelevant IFC data loss or the exchange should be limited to partial models that contain application-relevant IFC data. The latter seems to be the most practical solution, otherwise software developers will face high levels of data redundancy. Moreover, they will have to maintain both, the coherency of their own data as well as the data produced from other applications processing the same model. Among the examples for using IFCs in real projects are: The Headquarters for the Danish Broadcasting Corpora-

tion (Karlshøj, 2002), LBNL E-Lab Building (Bazjanac, 2002) and one of the most important experiences of designing real projects where IFCs have been implemented for interoperability reasons is the Helsinki University of Technology Auditorium Hall (HUT 600), (Kam et al, 2002) in Finland. In this example IFC-based data exchange took place among architects, mechanical engineers, construction managers and 4D research collaborators using IFC release 1.5.1. The project team exchanged architectural models, thermal simulation data, mechanical component geometries, building components and material data.

The design team reported that they managed to save 50% of the design time by minimizing data re-entry. Thermal simulations and cost estimation could immediately and directly make use of the IFC model. It enabled interoperability between 3D geometric and non-geometric data such as thermal values, construction material and assembly properties.

On the other hand, they experienced some geometrical misrepresentation by middleware and software, loss of object information, confusion in interdisciplinary revisions, large file sizes and absence of specific applications requirements. In reporting such experiences, the partial data exchange and the support of interdisciplinary revisions were given a top priority among the identified short comings.

The project's report also emphasised that partial model exchange will allow each discipline to read data that is pertinent to it. This should result in reducing time and the burden to import the entire IFC model. Furthermore, partial data exchange should have the potential of minimizing the risks of erasing or corrupting other idle project data. IFC model servers were identified as a potential solution for partial model exchange problems.

2.3 Model views and management of exchange scenarios

A drawback of some previous research efforts might have been that ICT technologies used to sit in the driver seat and steer partial model exchange scenarios. However, there is a great need to understand the connections to a larger context, where the end user's value chain requirements and procurement systems' demands are the driving factors, i.e. research efforts should be driven by end users' needs rather than ICT solutions.

2.3.1 Model views

The IAI has published the IFC Model View Definition Format in year 2006 (Hietanen, 2006). This report includes the main procedures that should be followed in order to reach a Model View Definition. It presents a road map that identifies the main processes that should be followed all over the lifecycle of a Model View Definition. It seems that the IAI has recently become aware of the importance of mapping business processes and data exchange requirements. They are key enablers for allowing deployment of new technologies to take place. This would ultimately increase the number of end users and consequently speed up the evolution and maturity of such technologies. This is particularly true if we consider the amount of end users' (millions) efforts that can be con-

tributed to the development process in comparison to the small group of IAI/ IFC experts alone.

2.3.2 Partial model exchange

According to (Lockley et al, 2000) partial model exchange takes place in two different forms:

- Schema oriented partial exchange
- Instance oriented partial exchange

2.3.2.1 Schema oriented partial exchange

The partial schema defines a data subset of the core schema. This data subset represents the relevant data objects' requirements of a specific application or activity. An example of this approach is the exchange of all walls of a building regardless the included openings, or the exchange of quantities regardless the objects' geometry. In this approach it is emphasised that the subset schema is part of the main schema.

Entities' instances are extracted "*Checked Out*" from the central repository in a long term transaction in the form of a STEP-P21 file, that is used by the importing application, i.e. a splitting process. Most probably the extracted partial model will have to be mapped to the internal data structure of the importing application. Information loss depends on the relevance of the imported partial model's objects to the internal data structure of the application.

The above splitting or partial model extraction can take place in two forms. (1) Cutting the extracted instances away from the model. (2) Keeping the original model as it is and extracting a copy (clone) of the required instances. The second approach is safer. However, it requires more cautious mechanisms in collaborative working (e.g. locking) and also at later stages, when a merge of the modified sub-model is required, otherwise the consistency of the model is at risk.

After accomplishing the required tasks by the importing application, the model is ("*Checked In*") reintegrated to the original core model through a merging process. Experiences from the COMBINE research project (Augenbroe, 1995) showed that there is a need for an output sub-schema that is different from the input sub-schema. The use of sub-schemata is mainly aimed at avoiding the product model being corrupted by inconsistent data. Thus, sub-model instances have to be validated against the EXPRESS rules before being transferred, in order to ensure coherence.

The above scenarios can be carried out directly by using the EXPRESS-X (ISO 10303-P14, 2005) mapping language. EXPRESS-X mapping schemata are written to manage data transfer from the core model to the sub-models. In this case, it will be a copying schema that defines the model's subset. In other cases, it can be a mapping schema that maps the subset to an entirely different EXPRESS schema (a *Business Object*, can be a non-IFC schema). An advantage of this copying approach is that it preserves objects' Global Unique Identifiers (GUID) for object instances that are not newly created, and hence the identification of project instances in the merging (integration) and comparison processes is guaranteed for all instances that possess a GUID (subtypes of IfcRoot).

3 SOLUTION CONCEPT (FIOPE)

The Flexible Instance Oriented Partial Exchange Environment (FIOPE) is a new approach that depends on an instance oriented approach rather than a schema oriented partial exchange scenario. The following sections discuss the approach in addition to an implementation example for an IFC 2X model:

3.1 End user involvement

As a result of the discussion in section 2.3.1, it could be concluded that the involvement of end users in the development of partial exchange requirements and mapping their processes is the key to any successful deployment.

End user's are not considered to be software developers only. Architects, engineers, all stake holders and value chain members are considered to be end users as well. In order to involve end users in the development / deployment process of IFC partial model exchange, it is totally impractical to think of educating them the IFC/EXPRESS, EXPRESS-X or SDAI technologies to be able to create a sub-schema to respond to their requirements and map their processes. It is envisaged that end users should have ready made flexible user friendly simple tools that enable them to define their partial exchange scenarios. Meanwhile, the available EXPRESS/STEP tool boxes (e.g. EDM (EPM, 2004), (EuroSTEP,2007), (STEPTools,2007)) are too complex to be used by actors who do not have any idea about the underlying technologies.

An important end user is the project manager or the actor who controls data acquisition and management. He is the one who controls partial model distribution among AEC/FM disciplines at each phase of the construction project and also at transferring the project data from one phase to the other (e.g. from design to construction). Nevertheless, this depends on the selected procurement route and the degree of integration between design and construction activities (e.g. whether it is a turn key project, design and build, BOOT, PFI and so forth). Moreover, in teamwork collaborative environments, team members need to share partial models among themselves, regardless the software application they are using. Consequently, there is a need for a degree of freedom to be able to define the partial exchange contents themselves. Usually, the partial exchange takes place in three dimensions: (1) Is among team members of the same discipline, (2) Among multiple disciplines (3) From one stage of the project to the other. At this stage of the research work, the paper focuses only on the 1st dimension.

3.2 FIOPE specifications

The FIOP approach enables the end user to carry out IFC partial exchange operations without the complications of both the partial EXPRESS schema generation or SDAI. The prototype is currently under development on the basis of the IFC2X3 model version. The main challenge in this approach is to achieve consistency and avoid data redundancy on one side, and from the other side to provide a flexible user friendly environment for the wide spectrum of end users. These two aspects (flexibility of use and

data redundancy) lie on two opposite ends of a continuum. This approach is aimed at giving end users a chance to participate in developing and defining their own partial exchange scenarios. By evolution, trials and errors, the partial exchange requirements and technological demands will force themselves to exist. At this point, well defined EXPRESS sub-schemata can begin to have some meaning for the end user who is not deeply involved in the underlying technologies.

3.2.1 IFC model splitting

One of the main functionalities needed by the project manager or the body that is responsible for data acquisition is to be able to create sub-models and distribute them among different actors. In the schema oriented approach, this would have been done by copying the elements using an EXPRESS-X mapping schema from the source model to a target model. This necessitates that both the mapping schema and the partial model schema should exist in advance. The FIOPE approach, on the contrary, gives the chance to the end user to determine him/herself on the spot the elements that should be packed to the partial model in a Pack & Go - Drag & Drop scenario. The latter necessitates the existence of a 2D/3D IFC CAD viewer, the IFC model tree spatial structure in addition to filtering and selection functionalities. A main challenge that faces FIOPE is keeping the consistency of the IFC model and avoiding data redundancy at the same time. Accordingly, it is very important to identify places in the IFC model schema structure, where it is possible to make a cut in the model. It resembles carrying out a surgical operation. By observing the EXPRESS-ISO 10303-P11 definition of the IFC model, two main places for cutting/splitting the sub-model from the parent model could be identified as shown in figure 1. (1) Optional attributes, (2) Inverse Attributes.

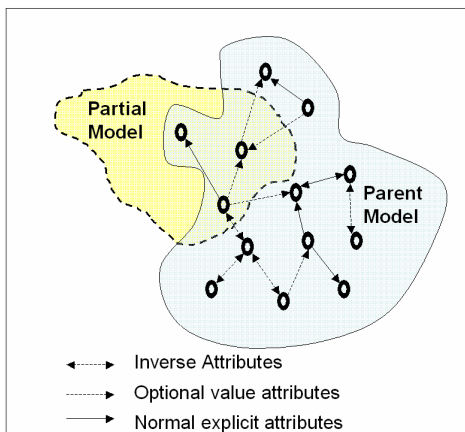


Figure 1. The relation between the partial and parent models with potential splitting edges.

3.2.1.1 Splitting at optional attributes

An attribute being specified as *Optional* means that it can be assigned a null value. This is represented as a “\$” in the STEP-P21 exchange format. In EXPRESS, (Schneck et al, 1994) state that “*Unless an explicit attribute is declared as optional valued, a legal value is mandatory.*”. Saying that an entity has an Optional attribute does not mean that the attribute itself is optional. It means that the value of the attribute is optional, i.e. the value is not always needed, but the attribute itself has to exist. It should

also be mentioned that optional attributes do not change the interpretation or the behaviour of the entity (ibid). Consequently, cutting or splitting the IFC model at this place should not affect the model’s coherence. It does not mean a physical cutting to the model. It means copying the element of the model with the optional attributes given the value null to prevent pulling (copying) the entire tree of referenced objects. This should be decided by the user through a graphical user interface.

3.2.1.2 Splitting at inverse attributes

Another suitable place for splitting/cutting the sub-model from its parent model while preserving the model’s coherency is at the Inverse Attributes for the following reasons:

1. Inverse Attributes are not mapped to the exchange STEP-P21 format, i.e. not included in the IFC/STEP files.
2. The role of Inverse Attributes in EXPRESS is to show existential dependence rather than the definition of the entity itself (Schneck et al, 1994).
3. By observing the use of Inverse Attributes in the IFC EXPRESS definition schema, it could be noticed that they are in most of the cases linked to a relation instance (IFCRELxxx) that links an element with other elements of the model, (e.g. A building storey to its constituents). This relation plays the role of a cross reference table in a relational database. Moreover, the reference between the abstract entity *IfcObject* and all its subtypes can be restored at reintegration (merging) time.

Due to the above reasons, the Inverse attributes are considered as a suitable place to perform a cutting when splitting a sub-model.

The splitting process also copies the context of the selected elements together with the non-optional object instances referenced by their attributes (e.g. *IfcOwnerHistory*, that contains a lot of information about the creator of the object, the software application, version number and so forth). The importing applications will not be able to process the IFC model without contextual information like the *IfcProject* instance, the definitions of measuring units and so forth. Therefore, there is a minimal context that should be copied with the selected elements to the sub-model. This context is also needed later at the reintegration stage.

3.2.2 IFC model integration / merging

After the sub-model has been worked upon by software applications, it has to be reintegrated to the parent model. The main challenge in the integration process is the *comparison* between the elements in the sub-model and their existing counterparts (old copy) in the parent model. The comparisons can be differentiated into two types: (1) *Shallow Comparison* and (2) *Deep Comparison*.

3.2.2.1 Shallow comparison

In the shallow comparison process, the primitive attributes of the IFC elements are compared. References are not followed. This comparison identifies the elements counterparts in both the sub-model and the parent model through GUIDS (Global Unique Identifiers). Consequently, the rest of the primitive attributes are compared. The first step in this comparison process is to make sure

that the same model is being compared. This is done by insuring that both models possess the same *IfcProject*¹ instance with the same GUID. It is important here to mention that in primitives comparison like (*double*, *REAL*, *float*) a tolerance value should be specified, otherwise values will be shown as different, even if they are equal to the 3rd or 4th decimal place. This tolerance is defined in the *IfcProject* context. This kind of comparison can pick up changes in the values of primitives like the width or height of an *IfcDoor* or *IfcWindow*.

3.2.2.2 Deep comparison

The deep comparison on the contrary, follows up the attributes' references and compares an entire tree structure. It is not a direct or a straight forward task. The tree structure of the sub-model is different from the parent model. It can include new elements that do not exist in the main model or miss elements that exist in the parent model. Elements from the original model could have been erased. Comparing counterparts in both models according to GUIDs or position in the tree does not satisfy the comparison's needs. This is attributed to the fact that only IFC elements that are derived from *IfcRoot* possess a GUID. For example a material (*IfcMaterial*) that is linked to a wall through the relationship (*IfcRelAssociatesMaterial*) does not possess a GUID. Hence, the only means of comparison is by comparing the type of the attribute and its string value. It is even worse when we consider geometry comparison. All elements that belong to the geometry resource schema at the resources level of the IFC model do not possess a GUID. Hence, if the values of the primitives are different, then the situation is given back to the user to handle it himself by either confirming or rejecting the new version of the geometry as a whole. Conflict detection in 3D is a matter that is expected to be resolved by the user's intervention through the graphical user interface.

3.2.2.3 The merging process

In the merging process, the elements of the partial model are copied to replace their original counterparts in the parent model. At this point, all optional attributes possessed by any IFC element in the parent model that are absent in the partial/sub model are re-instantiated. Thus, all links to the elements that are not part of the partial model are retained.

3.2.3 Needed tools

In order to be able to perform the operations defined in the above sections, a set of tools that enable the manipulation of the IFC model is needed. EXPRESS is NOT a programming language and hence, there is a need to bind the EXPRESS data structures to a programming language. Moreover, the STEP-P21 exchange format has to be read and interpreted by the manipulating application. It is envisaged that the end user will carry out all splitting and merging operations on a graphical user interface that visualizes both the IFC CAD view and the IFC model spatial tree structure (*IfcProject* → *IfcSite* → *IfcBuilding* → *IfcBuildingStorey* → ...). The needed tools are the following: (1) STEP-P21 Parser, (2) An IFC/EXPRESS Interpreter, (3) IFC viewer (CAD and Tree), (4) IFC/STEP writer. Due to the limited space in this paper, for a com-

plete description of how these tools can be built for scientific research purposes, the reader is advised to refer to (Nour et al, 2005)

3.2.4 Example

As a simple examples that illustrates the principles of the FIOPE approach, a CAD model of a building storey was initiated in ArchiCAD 9.0 (GraphiSoft, www.graphisoft.de), as shown in (figure 2(a)). The model was splitted according to the FIOPE approach and the *IfcProject* instance containing an *IfcSlab* was exported to another CAD software; ADT 3.3 (Autodesk, www.autodesk.com) as a STEP-P21 sub-model. Walls and openings were added to the sub-model, as shown in figure 2(b)). The modified sub-model was exported in the form of a STEP-P21 file and re-integrated to the original parent model, as shown in figure 2(c).

After integration of partial models, the user could still identify the history of each element (who created it, when, by which software application, versions and so forth). Normally, if a CAD software imports an IFC STEP file and re-exports it, the header file information and objects' owner histories are changed to contain information about the last processing application. In the above example, each action by each actor and each software application can be traced back to its doer.

The above process is not as simple as it seems to be. Each *IfcWallStandardCase* consists of eight attributes that are exchanged through the STEP-P21 file. Six of the eight attributes have optional values. The only two mandatory values are the GUID and the owner history that are inherited from the entity *IfcRoot*. Hence, the user is free to decide whether to pack the optional attributes and take them to the sub-model or not. In the case of a CAD application, the local placement and the shape representation of the object are essential. Therefore, they are copied to the partial model. Other optional attributes like the description, object type or tag of the wall are left behind.

The user has also the ability to choose between the inverse attributes to be copied to the partial model. In order to take the wall's material to the sub-model, the instance of the linking relation *IfcRelAssociates* has to be copied together with its referenced *IfcMaterialSelect* and its references to the partial model. (Figure 3) shows an EXPRESS-G diagram that clarifies this concept. The user is free to determine the splitting edge of the model according to the exchange needs. In (figure 3), the user is able to chose the attributes according to the above rules. In this case, the user decides that the material of the wall and the wall's description (optional and inverse attributes) are not needed by the data exchange requirements.

In both cases of optional and inverse attributes, where the value of the attribute is non-primitive (i.e. a reference), the entire tree structure of the attribute and its references are traced and copied to the sub-model. A more advanced algorithm could also decide whether to take the optional or inverse attributes of the followed references.

The merging or re-integration process highly depends on comparing the tree structure of both the parent and partial model. Newly instantiated elements are copied to the parent model. The new attribute values of the old elements replace the old ones and, thus, the object retains both its modified attributes as well as the old ones.

¹ *IfcProject* is a unique entity in every IFC model.

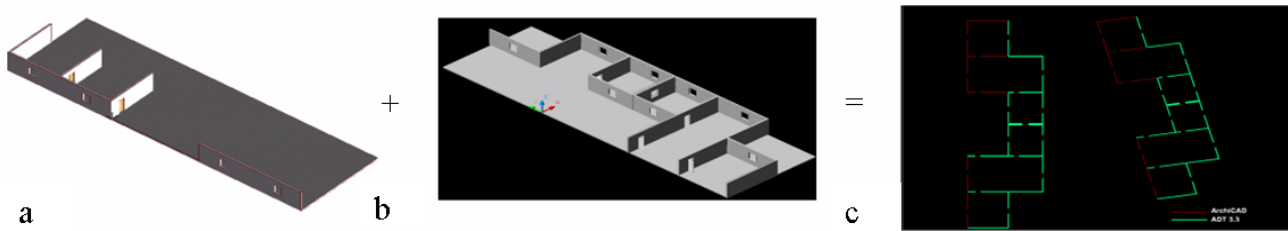


Figure 2. The integration of the partial model to the parent model. (a) is the parent model initiated in ArchiCAD, only the *IfcSlab* and *IfcProject* instances were exported. (b) The rest of the walls in the building storey were drawn in ADT 3.3. (c) The partial model “b” is merged to the parent model “a”.

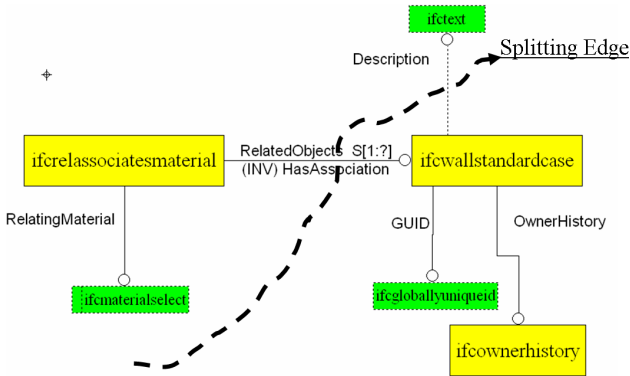


Figure 3. An EXPRESS-G diagram showing the splitting of the IFC model at inverse and optional attributes. The diagram is an abstraction from the author to enable the reader to follow up the references.

4 CONCLUSION

It is concluded from the discussions in this paper that the end user’s contribution is essential to the success of any process mapping that can be implemented later by a partial exchange schema. The end users are considered to be the wide spectrum of users who contribute to the project’s whole life cycle and value chain. This contribution will determine the process requirements and identify the industry needs. Consequently, partial exchange scenarios can be identified and developed by evolution rather than revolution.

The FIOPE approach is an instance based partial exchange approach that enables end users who are not EXPRESS, IFC or STEP experts to define their own partial exchange patterns on a user friendly graphical user interface.

The approach makes use of the optional and inverse attributes of the IFC model in splitting and re-integrating (merging) models. The comparison of IFC tree data structures is a crucial aspect to the success of this approach. A simple implementation example was demonstrated by using two different CAD packages for splitting and merging partial models containing few IFC elements.

The advantage of the developed approach is its simplicity and the ability of non-EXPRESS or STEP experts to use and define their own exchange content. However, the disadvantage of this freedom is that it is difficult to guarantee a high degree of data consistency and coherency in comparison to the schema oriented approach, where checks and validation against EXPRESS rules can be performed.

The developed approach is expected to give a great push to the technology pull, which would consequently help process re-engineering in various procurement systems in the AEC/FM domain.

ACKNOWLEDGEMENTS

“The result presented in this paper is part of InPro (<http://www.inpro-project.eu>) - an integrated project co-funded by the European Commission within the Sixth Framework Programme (2002-2006).”

REFERENCES

- Augenbroe, G (editor) (1995). COMBINE 2, Final Report, CEC-JOULE program, Brussels, http://erg.ucd.ie/combine/what_about.html
- Bazjanac, V. (2002). Building Information Modelling for the e-Lab at LBNL, <http://www.aia.org/SiteObjects/files/elab.pdf>
- EPM Technology (2004). Express Data Manager (EDM), <http://www.epmtech.jotne.com>
- EuroSTEP (2007). <http://www.eurostep.se>
- Hietanen, J. (2006). IFC Model View Definition Format, Version 1.0, International Alliance for Interoperability.
- IEEE (1990) Institute of Electrical and Electronics Engineering. IEEE Standard Computer Dictionary. A compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.
- ISO 10303-11(1994). “Industrial automation systems and integration”-- Product data representation and exchange-- part 11: Description methods: The EXPRESS language reference manual.
- ISO 10303-21(2002). “Industrial automation systems and integration”-- Product data representation and exchange-- part 21: Implementation methods: Clear text encoding for exchange structure.
- ISO 10303-14(2005). “Industrial automation systems and integration” -- Product data representation and exchange -- Part 14: Description methods: The EXPRESS-X language reference manual.
- ISO/PAS 16739 (2005). “Industry Foundation Classes, Release 2X, Platform Specification”, (available at <http://iso.nocrew.org>)
- Kam, C. and Fischer, M. (2002). Product Model & 4D CAD – Final Report. CIFE Technical Report TR 143, Stanford University, available at <http://www.stanford.edu/group/4D/download/c1.html>
- Meriam-Webster Dictionary (2007). <http://www.m-w.com/dictionary/Interoperability>
- NCCTP glossary (2007). The Network for Construction Collaboration Technology Providers (available at <http://www.ncctp.net/glossary.htm>)

- Oxford Dictionaries (2007). AskOxford.com
http://www.askoxford.com/concise_oed/interoperable?view=uk
- Karlshøj, J. (2002). IFC Pilot Project – Headquarters for the Danish Broadcasting Corporation,
http://cic.vtt.fi/niai/IAI_Summit_2002.htm.
- Lockley, S. and Augenbroe, G. (2000). “Data Integration with Partial Exchange”, Proc. Of International Conference on Construction Information Technology, INCITE 2000, Hong Kong, pp 277-291
- Nour, M. and Beucke, K. (2005). “Manipulating IFC Model Data in Conjunction with CAD”, Proceedings of the 22nd Conference on Information Technology in Construction CIB W78, Dresden. Available from Construction Informatics Digital Library <http://itc.scix.net>
- Schneck, D. and Wilson, P. (1994). “Information Modelling: The Express Way”, Oxford University Press, New York, ISBN 0-19-508714-3
- STEP Tools (2007). STEP Tools Inc., www.steptools.com.

