RDF-BASED DISTRIBUTED FUNCTIONAL PART SPECIFICATIONS FOR THE FACILITATION OF SERVICE-BASED ARCHITECTURES

Jakob Beetz, Bauke de Vries, Jos van Leeuwen
Eindhoven University of Technology, Design Systems group, the Netherlands

ABSTRACT: In this paper we highlight research and development that is done in the larger context of a service oriented architecture framework for the support of design decisions. We are going to illustrate how methods that adhere to the "open world assumption" (OWA) can be used to construct semantically meaningful information fragments from larger models. We are demonstrating the composition and use of Functional Parts specifications as RDFS graph patterns. We outline a prototype that applies RDF(S) sub graph extraction and merging with queries and rules in distributed scenarios using models based on the IFCs that have been notated as partitioned OWL models. We are showing how these sub graphs can be used as machine-readable information exchange requirements not only for existing models but also for the semi-automated integration of newly added conceptual models as project-specific augmentations.

KEYWORDS: building information models, semantic web, service oriented architectures, IFC.

1 INTRODUCTION

The uptake of Building Information Models in industry – although accelerated significantly over the last couple of years – is inhibited partly by factors that are rooted in the complexity of the models and a lack of rigid methodologies to deal with this complexity. The current IFC model consists of over 600 entities, more than 300 types and some 1.500 attributes, not counting the potential growths of schematic information in actual model instances that are extended with custom PropertySets. To allow users to reduce this complexity some aspects of the past milestone researches conducted in the BIM area, such as the GARM, and the RATAS efforts introduced mechanisms for cascading abstractions and aggregation levels and numerous authors have indicated the necessity of views and partial models from the early days of onwards (Amor et al 1992). A recent development aiming – among other things – at the specification of model views for the IFCs for specific information exchange tasks is the Information Delivery Manual and Framework (IDM). However, partly due to the inherent technological underpinnings of the time, all these approaches share a common methodological issue that is related to the main technical means of modeling: The family of ISO STEP EXPRESS technologies. Although the creation and use of multiple schemas is included in EXPRESS' overall design, no rigid methods have been standardized that regulate the exchange of and interaction with schemas among distributed repositories in RDBMS environments. Secondly, its "closed world assumption" (CWA) wherein incomplete information results in errors (negation as failure), constitutes a considerable obstacle in dynamic specification, extraction and merging of sub-models. A third area that can be identified as one of the most urgent issues for a successful adoption of BIM-based information exchange is the general lack of low-cost tools to operate on such data.

2 RELATED WORK

2.1 *Functional parts, views and partial models*

As part of the work for the General AEC Reference Model (GARM) Gielingh (1988) introduced a conceptual modeling framework that included the decomposition of complex objects into Functional Units (FU) and Technical Solutions (TS) which became known as "Gielinghs' Hamburger Model". In this approach the conceptual characteristics of an artifact – the functions it has to perform and the requirements it has to fulfill – are specified separately in an earlier stage. Actual implementations of these functions can then be added in later stages or changed without affecting the overall composition.

A similar adapter approach – albeit with a different target and focus – was chosen for some aspects of the Information Delivery Manual (IDM) effort, which is currently developed mainly by Jeffrey Wix and Kjetil Espedokken (see Wix (2005)). The IDM effort aims at establishing a commonly agreed methodology to specify and standardize building industry related business processes and the information that has to be exchanged during these processes. The three main components to establish such a standard are workflow and information flow charting techniques called "Process Maps" (PM), informal yet structured descriptions of information I/O referred to as "Exchange Requirements" (ER) and a more technical aspect of information modeling that is in the focus of the work presented here: the "Functional Part" (FP). The latter component aims at standardizing the concrete information snippets that have to be handed over by software implementations that support the methodology. Its most important aspect is to specify which is the bare minimal information that a software artifact has to provide as output or can expect as input and what optional additional data is to be expected. Practically this is achieved by formulating data schemas that constitute sub-models of the Industry Foundation Classes (IFC). A side aim of the creation of small model chunks is a reduction of complexity that the overall model brings which currently constitutes a severe entry threshold for software vendors.

In our own research, the need of such formalized information exchange interfaces between software entities differs from this business oriented perspective. Yet we believe that the work we have done to address these issues could be applied in the above mentioned context. Similar approaches based on graph-theory have been suggested earlier by Luiten et al (1998) and the facilitation of Semantic Web technologies for the Product and Building Model sector receives increasing attention by various research works such as the InteliGrid (Turk et al 2004) and SWOP (Böhms et al 2007) projects.

2.2 *An approach based on distributed knowledge models*

A central idea in the approach we are proposing is the facilitation of rigid logical knowledge modeling methodologies for the description AEC information. Based on the fusion of two different families of information engineering – frame systems and description logic – the ongoing Semantic Web (SW) effort has led to the standardization of powerful methods and technologies to describe domain knowledge in a semantic machine-interpretable manner. The foundation of our suggested adaptation of these technologies for the AEC sector is the conversion of the commonly accepted standard to describe building information, the IFC model, to a distributed knowledge model defined using SW technology. Despite the fact that its purpose and aim is the exchange of information between applications, we are convinced that its extensive description of much of the relevant information can serve as an excellent basis for a knowledge model that can be extended with project- and company specific information and rules. Early stages of this adaptation work have been covered in parts in Beetz, van Leeuwen and de Vries (2005) and Beetz et al. (2006).

2.3 *Distribution of schemas*

One of the most important aspects of this earlier work is that the generated OWL model is based on of the Resource Description Framework (RDF see Lassila and Swick (1998)). This framework (whose most prominent use today is the Really Simple Syndication (RSS) of web pages) allows the composition of large graphs from object-predicate-subject triplets p(O,S) whereby each one of the three components is considered a resource which may reside anyplace that is identifiable by a URI. For the purpose of maintaining large schemas and populations of them, as is necessary in BIM environments that deal with a multitude of specialized domain models, this has several advantages: Instead of having a single huge monolithic schema and several scattered extension snippets (as is the case with the normative Property Set (PSet) extensions) schemas can be consistently separated and assembled into thematic categories from the beginning onwards[1]. Although language features exist for multi-schema constructs and their mapping in the “natural” schema language of the IFCs – the STEP EXPRESS family – the support and use of these possibilities is limited at present.

2.4 *Distribution of instance populations*

The distributed nature of the underlying RDF framework is even more important when it comes to populations of schemas and the creation of partial model views from such populations. A very serious conceptual limitation of STEP part 21 populations is the fact that it is a simple ordered collection of individuals who’s only indexing and identification mechanism is an integer value that is unique only within a single population file. Secondly all attributes are order-dependent lists attached to the entity. The drawback of this efficient and concise serialization optimized for minimal file sized is the work necessary to resolve the semantics. As for the IFC model, only a fraction of all information has an extra attached unique id that is valid in a global context across population file borders (the “GUID” STRING attribute of IfcRoot and its descendants). In practice this means that modifications such as insertions or deletions to some parts of the overall population model are very likely to affect the order of all other entity instances and hence the ability to external information snippets to reference them. For the extraction and – more importantly – merging of partial model views this renders many consistency problems unsolvable without cumbersome and error-prone extra bookkeeping. In RDF on the other hand, every component of a triplet p(O,S) has its own ID that is valid across system borders. This makes it possible to reference schema or population elements. For the concrete case of partial model views and the concept of Functional Parts this eliminates the necessity to replicate schemas and instances as is the current practice in the IDM.

3 DEFINITION, CREATION AND VALIDATION OF FPS WITH OWL/RDF GRAPHS

3.1 *A concrete example*

To illustrate our approach, the following simple example is considered: A Decision Support application requires some thermal transmittance U-values for windows of a building as input. To keep the example small and concise, a small FP “fp_thermalWindow” is constructed that is needed as the minimal required input for the application. In our overall collaboration framework, this decision support application is represented by a wrapper agent that facilitates the communication with other applications.

---

[1] In our implemented prototype we have used the domain separations of the model as categorization for partial schemas that reside in separate xml namespaces in their own respective files such as ”ifckernel.owl”, “ifcsharedbuildingelments.owl” and “ifchvacdomain.owl”

3.2 *FP definition with RDF query languages*

The formulation of such FP by means of RDF can be accomplished in two general ways:

- A self-contained replication or reformulation of all relevant entity definitions and their inheritance trees including the respective attributes as independent partial schema that is completely decoupled from the original IFC model schema.
- Compilation of references to the according entity and attribute definition in the corresponding model schemas

While technically feasible, the first approach - which is a direct adaptation of the current IDM approach - does not overcome the weaknesses with regard to semantic coherence.

The advantages that RDF brings in this regard over the traditional STEP/EXPRESS methods become apparent in the second approach: By pointing to the corresponding (distributed) schemas, an OWL/RDF - aware application 'knows' the semantic definition of a window and its properties by pulling the defining triplets from the schema resources when their availability becomes necessary. Moreover, the actual expansion of the complete definition of the IFCWindow class is not necessary for simple operations such as partial graph extraction, since the equality of the resource URIs (which function as UIDs) is a sufficient comparator for a processor to e.g. extract a "thermal windows" view from a large model. To look for and extract the relevant sub-graph we have to formulate a graph pattern to search in our original model. For the example at hand, the required sub-graph can be formulated as a digraph

$$G = (\{n_{win}, n_{rel}, n_{prop}, n_{val}\}, \{(n_{win}, n_{rel}), (n_{rel}, n_{prop}), (n_{prop}, n_{val})\})$$

with

n_{win}= IfcWindow, n_{rel}= IfcRelDefinesByProperties,

n_{prop}= IfcPropertySet, n_{val}=IfcPropertySingleValue

3.3 *Partial model / view extraction*

Using the FP graph we formulated above, we can generate a small graph pattern matching query in one of the languages such as SPARQL, SeQRL, RQL, etc. for which some fast and efficient FOSS implementations exist and pull a fraction from a large model (which itself can be distributed over various locations) as illustrated in figure 1 . Using graph query languages to operate on large models like average IFC models for theses simple tasks is less complex than the use of full-blown rule and reasoning engines (whose use we will illustrate later for the semantic validation of our sub models). In cases where the standardized query operations do not suffice, many implementations allow the creation of domain specific extensions. One of such useful extensions could be, e.g., the implementation of spatial operators such as currently worked on by Borrmann, van Treeck and Rank (2006)

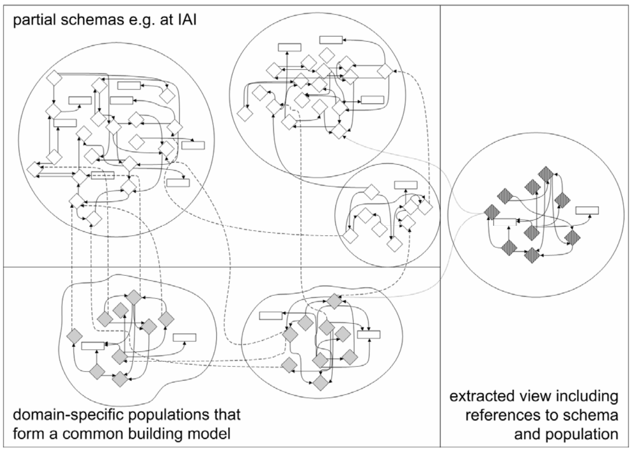

Figure 1. Partial view extraction of distributed RDF graphs and schemas. Cross-references between schema and population files or DBs are being carried over into the extracted submodel.

For the concrete example we make use of the SPARQL 'CONSTRUCT' feature, which allows the creation of new graphs. We extract a sub graph containing all windows that have properties attached via IfcPropertySets of an original (distributed) model by

```
CONSTRUCT {
          ?window a ifcsharedbldgelements:IfcWindow .
          ?window ifckernel:IsDefinedBy ?defines .
          ?defines a ifckernel:IfcRelDefinesByProperties .
          ?defines ifckernel:RelatingPropertyDefinition ?propDef .
          ?propDef a ifckernel:IfcPropertySet .
          ?propDef ifckernel:HasProperties ?singleProp.
          ?singleProp a ifcpropertyresource:IfcPropertySingleValue .
          ?singleProp ifcpropertyresource:Name ?name .
          ?singleProp  ifcpropertyresource:NominalValue ?value.
}
WHERE {
          ?window a ifcsharedbldgelements:IfcWindow .
          ?window ifckernel:IsDefinedBy ?defines .
          ?defines a ifckernel:IfcRelDefinesByProperties .
          ?defines ifckernel:RelatingPropertyDefinition ?propDef .
          ?propDef ifckernel:HasProperties ?singleProp .
          ?singleProp ifcpropertyresource:Name "ThermalTransmittance".
          ?singleProp ifcpropertyresource:NominalValue ?value.
}[2]
```

This results in a partial graph depicted in Figure 2 that only contains the minimal information needed by the target application. At the same time, the graph carries provenance information by pointing to the corresponding schema elements and occurrences.

To keep the view consistent with the global model, the target application could re-evaluate the slot filler values by resolving the URIs. However, for this to work additional version management over time has to be done. Several approaches for temporal logic and provenance data in RDF for the purpose of journaling and model consistency are introduced by Gutierrez et al (2005), Futrelle (2006), and Huang and Stuckenschmidt (2005).

---

[2] Note that it is implied that inferred symmetric properties owl:inverseOf have been asserted into the graph beforehand. isDefinedBy in this case has to be explicitly added finding the symmetric closure on the RelatedObjects property that has the domain IfcRelDefinesByProperties class.

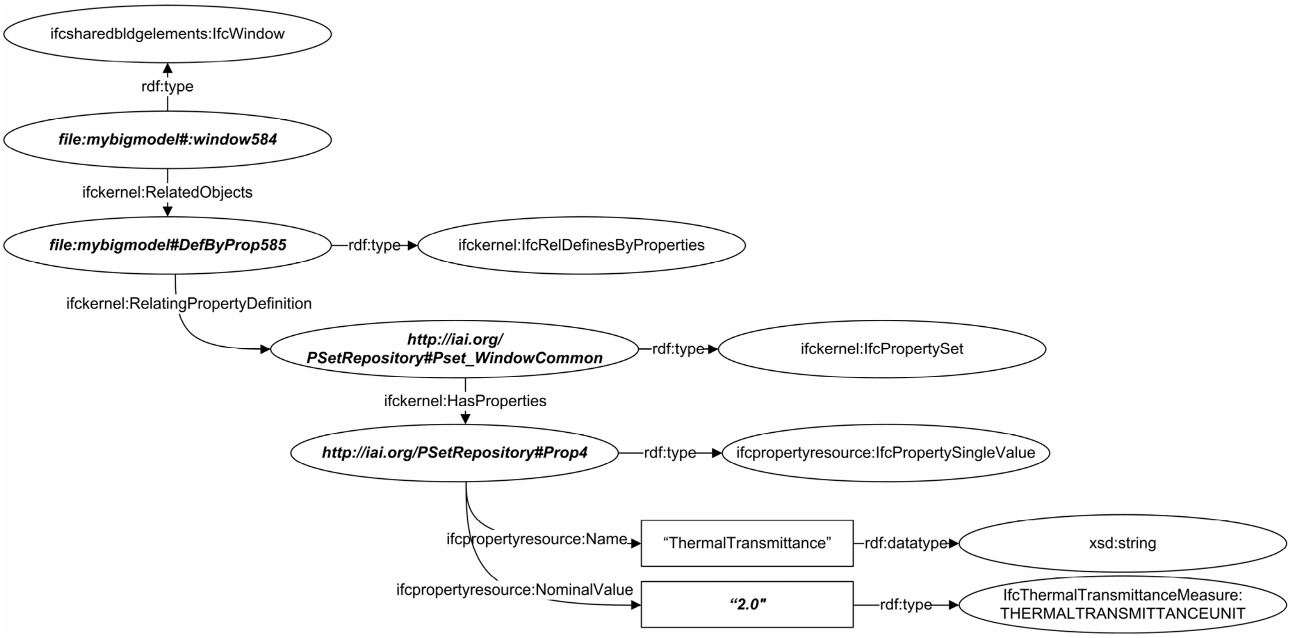

Figure 2. Minimal partial graph extracted from model graph.

3.4 *Validation using reasoning under open world assumptions*

A validity check for an FP for a window instance required for the above scenario can be formulated as Description Logic axioms (abbreviated):

$$\begin{array}{ll}
\text{fp\_thermalWindow} \equiv & \\
& \text{IfcWindow} \\
\cap & \exists\ \textit{IsDefinedBy}.(\\
& \quad \exists\ \textit{RelatingPropertyDefinition}.(\\
& \qquad \exists\ \textit{HasProperties}.(\\
& \qquad\qquad (\ni \textit{Name} \text{ ``ThermalTransmittance''}) \\
& \qquad \cap \quad (\geq 1\ \textit{NominalValue}) \\
& \qquad \cap \quad (\forall\ \textit{Unit}.\ \text{IFCTHERMALTRANSMSR}) \\
& \qquad\qquad) \\
& \qquad) \\
&)
\end{array}$$

This construct can be read as :"A ThermalWindow is a window that has some related property definition which has some property by the name of 'ThermalTransmittance'and at least one value 'NominalValue' all of whose Unit types are 'ThermalTransmittanceMeasures' ". It can be easily translated into concrete OWL syntax making 'fp_thermalWindow' an owl:equivivalenClass for which generic 'of-the-shelf' reasoners can find instance occurrences within the extracted submodel. To enable a reasoning engine to successfully find entailments of these nested axioms and assert them into the graph, the definitions of all participating concepts have to be known. This adds an additional level of complexity on top of the simple queries in the earlier extraction. While the simple queries are handled purely on the RDF graph layer, the semantic meaning of classes, their attributes and relations have to be known during the validation stage. During the inclusion of the necessary sub schemas - which is a standard operation for many existing OWL/RDF processor implementations – the only obstacle is some extra bookkeeping to handle cyclic references.

It might be considered a drawback that the Open World Assumption, which is the basis of all reasoning on OWL, does not allow us to extract all windows that do *not* have a 'ThermalTransmittance' value directly. In the Semantic Web world all information is considered incomplete (there might be some value for ThermalTransmittance that is not accessible in the current context) hence an answer to the negation of the second part of the above axioms returns 'unknown' rather than 'false'. However, in a scenario where we would like to be able to detect these (e.g., in order to prompt the user to fill in the necessary missing values), we could simply iterate over all the windows and mark those that have not been classified as fp_ThermalWindow for further processing (falling back to the level of conventional imperative programming).

4 PROTOTYPE IMPLEMENTATION

In a prototypical implementation of our system making use of FP formulated in RDF, we have created a GUI application that supports developers to assemble FPs from the partial schemas of the ifcOWL model we have presented earlier. In a similar fashion like the tool that Lee et al (2006) have implemented for the support of the GTPPM method to generate STEP sub-models a developer is able to select classes from the IFCs and their attributes to be included in the FP. A corresponding SPARQL graph pattern matching query is constructed, that is able to generate partial views from an ifcOWL file using a generic SPARQL processor such as the ARQ implementation in the Jena framework. Together with the semantic validation formulated by the owl:equivalentClass axioms (which arguably require some careful manual work at present but might be automated to a certain degree in future) these queries form the basis of a thin agent layer that is wrapped around a Decision Support tool. A skeleton agent is generated from an existing generic template that takes care of the basic communication within a bigger society of agents. The 'specialization' of the agent, its behavior and exchange of concrete and practical information is then specified on a pure content-centric meta-level. Not only can the agent state which input is required (fp_thermalWindows), it can also ask a model managing service to create the partial models by handing over the generated extractor code and receiving the view in return. This not only spares a developer of such application from dealing with large population models and unrelated schemas in small applications, it also saves a lot of unnecessary data traffic.

5 DISSCUSSION

The approach we have presented here has several advantages for developers of specialized tools in the building sector. Dealing only with the relevant fractions of a large building model reduces the work that is necessary to deal with complex BIMs. The partial models generated with the method introduced can be treated on different levels: On a pure XML/RDF level it is quite easy to process the information generated with one of the many existing low-level processors and extract information with XML Schema datatypes to map it against the internal model of an application. On the higher semantic level of OWL, rich and logic-based type information about the classes and their relations involved is available and can be combined with external ontologies and rules to create complex systems of small specialized applications.

The use of the RDF-encoded OWL to describe distributed building models adds a layer of computational complexity that might be considered inefficient for those large portions of a building model that describes geometry: For the storage and exchange of huge nested BREP and CSG structures semantic capabilities do not additional value as long as there are no specific algorithms to support logical reasoning on a geometric and topological level. However, we believe that in those regards that separate pure geometry-centric exchange models from BIMs enriched with meta-data attributing and specifying the components the addition of a logical level is very promising. Although the build-in distribution capabilities of the RDF stack do not solve all consistency issues ‘out-of-the-box’ we regard it as a promising starting point that can help to improve the use of BIMs in heterogeneous environments.

It might be argued that the cognitive threshold the Semantic Web stack introduces into BIM-centered operations even outweighs that of STEP/EXPRESS. We believe that the large amount of ongoing work in various research areas and industry fields that involves Semantic Web methods and technologies will make it easier to deal with this kind of information in the long run. We think that the availability of industry-scale and free tools (including persistency frameworks) is of great use to enable the uptake of flexible, distributed and dynamic BIMs especially for small businesses and research institutions.

6 CONCLUSIONS

In this paper we have presented a novel method to create partial model views from RDF-encoded IFC models by using SPARQL queries. We have shown how semantic information of the underlying OWL models is preserved by references and how this method has significant consistency advantages over traditional STEP/EXPRESS approaches. A validation method of the generated partial models has been outlined suggesting the use of available reasoning algorithms for the classification of model views as Functional Parts. We have argued that the use of theorem proofing algorithms can serve as a rigid basis for the facilitation of distributed building information models in heterogeneous environments. We have described a prototypical implementation of a GUI tool that supports developers in creating queries for the extraction of Functional Part model views. We have outlined how the suggested approach can help to decrease the work necessary to integrate specialized tools into heterogeneous BIM-centered collaboration settings.

With regard to future work, we are especially interested to further investigate possibilities to semi-automize the creation of partial model validators with DL and rule systems. We are looking forward to apply our developments to real-world scenarios and to investigate possibilities for the integration into other frameworks such as developed in the InteliGrid project.

AKNOWLEDGEMENTS

Parts of the research presented here has been sponsored by the project IST-2004-004664 “InteliGrid” funded by the European Union.

REFERENCES

Amor, R.W. and Hosking, J.G. (1992) “Defining Multi-Disciplinary Views of a Computer Model of Buildings”, New Zealand Computer Science Research Students' Conference, Waikato University, Hamilton, New Zealand, 7-14.

Beetz J., van Leeuwen J.P., and de Vries B. (2005). “An Ontology Web Language notation of theIndustry Foundation Classes” 22nd CIB W78 Conference on Information Technology in Construction, CIB-W78, Dresden, Germany, 193-198

Beetz J., van Leeuwen J.P., and de Vries B. (2006). “Towards a Topological Reasoning Service for IFC-based Building Information Models in a Semantic Web context” In Rivard, H. et al (eds), Proceedings of the 11th International Conf. on Computing in Civil and Building Engineering ICCCBE-XI, Montreal, Canada, 3426-3435

Borrmann, A., van Treeck, C. and Rank, E. (2006) “Towards a 3D Spatial Query Language for Building Information Models” 11th Int. Conf. on Computing in Civil and Building Engineering, ICCCBE-XI. Montreal, Canada 2006. 1374-1385

Böhms, M. et al. (2007) “Semantic Web-based Open engineering Platform” EU STRP 016972 report D23 draft version available at http://www.swop-project.eu (last accessed 2007)

Futrelle J. (2006). “Harvesting RDF triples” Lecture Notes in Computer Science, 4145/2006, Springer, Berlin, Heidelberg, 64-72.

Wim Gielingh. (1988) “General AEC reference model (GARM) an aid for the integration of application specific product definition models”. Conceptual modelling of buildings, W74+W78 workshop, Lund, Sweden 165-178

Gutierrez C., Hurtado C., and Vaisman A. (2005). “Temporal RDF” Lecture Notes in Computer Science, 3532/2005, Springer, Berlin, Heidelberg, 93-107.

Huang Z. and Stuckenschmidt H. (2005). “Reasoning with multi-version Ontologies: A Temporal Logic Approach” In International Semantic Web Conference, 398-412.

International Organization for Standardization, (1994) ISO 10303-11:1994, Part 11: Description methods: The EXPRESS language reference manual

Lassila, O. and Swick, R. (1998) "Resource Description Framework (RDF) model and syntax specification" W3C Working Draft WD-rdf-syntax-19981008 online http://www.w3.org/TR/WD-rdf-syntax (last accessed 2007)

Lee G., Eastman, C. and Sacks R. (2006). “Generating IFC Views and Conformance Classes using GTPPM” ICCCBE. Montreal, Canada, 1715-1724

Luiten, G.T., Tolman, F. P. and Fischer, M. A. (1998) “Project-modelling in AEC to integrate design and construction”, Computers in Industry, 35(1), 13-29

Turk Ž., Dolenc M., Nabrzyski J., Katranuschkov P., Balaton E., Balder R., Hannus M. (2004) “Towards Engineering on the Grid”, ECPPM Conference, Istanbul, Turkey, 179-186

Wix J, (2005) “Information Delivery Manual – Using IFC to Build Smart” official project homepage http://www.iai.no/idm/learningpackage/idm_index.htm (last accessed 2007)