Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

Multi-level Ontology Mapping for a Cross-culture Collaborative Design

Gianfranco CARRARA, Antonio FIORAVANTI
Dept. Architettura e Urbanistica per l'Ingegneria – Sapienza Università di Roma,
Via Eudossiana 18, Rome, 00184, Italy, Tel: +39 06 44585165, Fax: +39 06 44585186
Email: gianfranco.carrara@uniroma1.it, antonio.fioravanti@uniroma1.it

Abstract: Design process is a very complex activity nowadays due to its cross-disciplinary nature, its delocalization, its new professional figures and its new performing requirements. Computer Supported Collaborative Work (CSCW) together with intelligent content and semantics are up-to-date methodologies to cope with these needs The research work aims at defining new methods, techniques and advanced ICT programs suitable for industries, professionals and technicians who, in various ways and with many specializations, operate in the field of the design and construction of complex building.

In the AEC sector designers (actors) often have difficulties in integrating their own work with others, as a deep collaboration implies working together on the same objects (components), and this can lead to misunderstandings, conflicts, incoherencies.

The purpose of this research has been to enhance, through suitable tools and methodologies together with the exercise of design collaboration, the overall quality of buildings. Such tools and methodologies are founded on correct mapping across ontologies of different cross-culture domains, so it is easier for actors to understand each other, for application programs to interface significant data and for design process to point out contradictory constraints.

The results of this research will be first be applied to provide university students in the schools of architecture and engineering with innovative tools for delocalized design learning enhanced by a cross-disciplinary collaboration.

Keywords: Collaborative Design, Ontologies, Knowledge Bases, Mapping

1. Introduction – Design and Collaboration

The general features of the methodologies and tools reported here have been conceived, developed, implemented, applied and tested on design of Architecture/ Engineering/ Construction (AEC) for the following reasons.

At present, design is a multidisciplinary e interdisciplinary complex activity that covers the entire range of the process leading from the conception to the construction of goods. This complexity has increased in modern times as a result of new needs and new problems, such as environmental sustainability, energy efficiency, new requirements and typology, new materials and products and sectoral regulations, which involve increasing specialization of the various professional figures involved.

The software and methods used today are unable to satisfy present-day needs as they date back to an earlier cultural generation, and are unsuitable for addressing today's challenges involving the global design of goods in their entirety and complexity.

The designers (from now on, actors [1]) actors involved in each phase of the process have to exchange information, knowledge and expertise, thereby fostering mutual understanding. To do this is necessary to have a common dictionary and to collaborate for more agreed goals.

In order to ensure their decisions are approached in such a way as to satisfy the requests/expectations of the various design actors, it is common practice to make use of meetings among actors by means of which relations of close collaboration are established among them.

Many forms of workflow have existed among actors involved in a design process, each of which presents advantages and limitations depending on its specific features and the context. These forms do not depend on the number of collaborating actors, but rather on other decisive features: timing of the actions (the overall time required for the operation and the presence or absence of phases), intersected competences, actor hierarchy, knowledge of the operating context and parallelism of the actions.

Regarding the intersection of competences in teamwork we define the following kinds of design processes.

Coordinate design is an elementary form of workflow where there are no intersections. That allows the design process to be broken down into juxtaposed phases. Information flow is essentially unidirectional and only minimal feedbacks happen; each processing phase begins only after its preceding ones have been completed: a process, a time. This type of interaction works correctly in well defined and steady working environment, which is generally intended to refine the details of the constructive design (Fig. 1).

Cooperative design is a more integrated form of design process where there are little intersections. All the actors strive to attain a common objective by working in a non linear, but pre-defined sequence, where each processing phase can begin even if not all the preceding ones have been carried out: many processes, a time. Just small quantities of information relating to the same entity are generally exchanged.

Collaborative design is the most integrated form of design workflow, in which actors' competences have more intersections, actors can help each other, gain greater insight into how their own work interacts with that of the others and how they can pursue their own objectives more effectively. Each processing phase can begin even if its preceding ones have not been completed, like in Cooperative design, but actors can also act on the same entities (components, concepts) very often exchanging large quantity of information relating to the same entity.

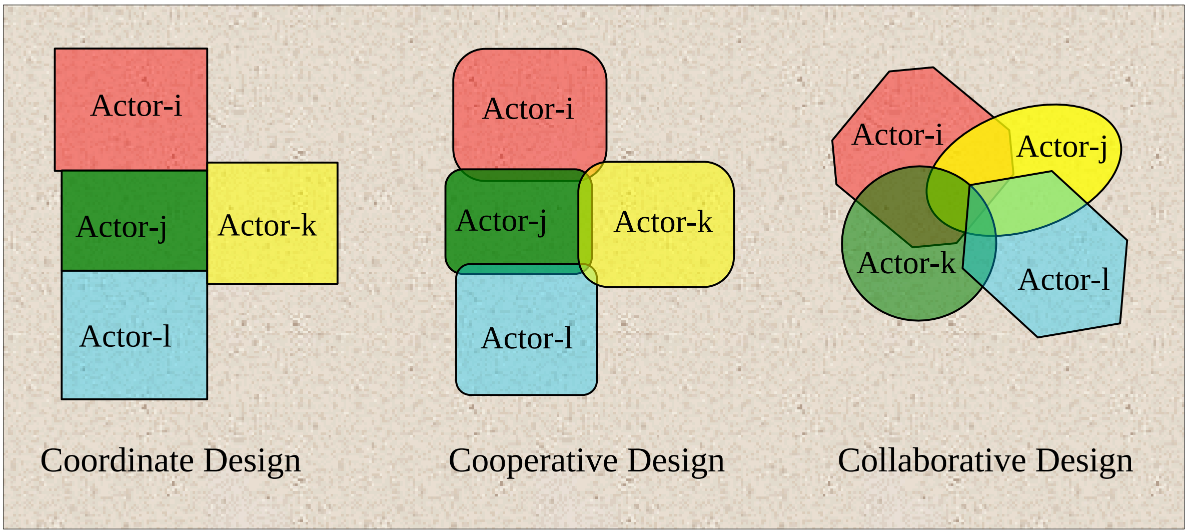

Figure 1 – Different types of teamwork

Collaboration in AEC design activity is suitable for this sector, as any building is a singular, integrated and complex system in a given context with interleaved problems. The global overall solution can be attained only by means of trade-offs among actors so that they can modify their own specialist goals and adapt their own specialist solutions.

The collaboration becomes all the more effective when it takes place at all the hierarchic levels into which a design and construction process is normally subdivided.

Collaboration can thus facilitate the discovery of new design solutions through the proactive contribution of each actor; it allows the various solutions by the different actors to converge towards a single overall solution; it helps the actors reciprocally to modify their own design solutions so that these ones can be more satisfactorily integrated into the overall solution; it allows the reciprocal malfunctions among the solutions proposed by the individual actors to be detected on time; lastly, it encourages the development of creativity through interaction among the different skills [2, 3, 4, 5].

To deal with Collaborative Design problems, software houses have proposed different solutions. In a not so distant past the main problem was to exchange data by means of neutral formats (standards): there were several initiatives such as PHIGS and STEP and so on, that can now be considered to be a small part-of the more general problem that is to find out how to formalize knowledge [6] in all its implications: technical knowledge + expertise + rules of thumbs + context-aware + physical lows, procedures, etc.

2. The State of the Art: BIM and IFC

For the past few decades it appeared that such standards and product models could serve the purpose of collaborative design, they failed as not flexible enough for design process. Moreover it is noticed that in the design process developed by usual application tools (various domain specific applications for analysis, simulation, verification, etc), actors have misunderstandings and lack of data.

In the AEC community several efforts have been devoted to overcome these difficulties in order to integrate competencies in a single application program and to share knowledge. Among the various initiatives, we mention *BIM* and *IFC*.

BIMs (Building Information Modeling) are a product models, recently driven by several CAD system vendors (Autodesk, GraphiSoft, Bentley, Nemetschek, etc.), which can describe the form (e.g., geometric information and its relationships) and attributes (e.g., physical characteristics) of a building throughout its life cycle. BIMs define a building with proprietary formats, their digital models are conceived with a top-down point of view, focus on components and not on the process, are a source of intelligent information about a building. To achieve a better interoperability between disciplines and software applications across the industry and professionals, that has become necessary and urgent, a pragmatic and efficacious basis for information sharing between different BIMs is essential.

For this purpose a second initiative with a different approach has been developed: the Industry Foundation Classes (IFC). It is an open XML standard (OOP) conceived with a bottom-up point of view and non-proprietary data model specifications, proposed by IAI – International Alliance for Interoperability. It is emerging very slowly among industries involved, due to his bottom-up approach. IFC aims at granting software interoperability, at exchanging more significant project data, so that nowadays CAD applications by major software houses, like aforementioned ones, can import and export their proprietary formatted files from/to IFC files. Such specifications represent a data structure supporting a digital project model, useful in sharing structured labelled (more understandable) data across applications, but they are neither intended for design needs nor for mutual understanding among actors, but mainly for (just!) production needs.

Until now exchanging contents among commercial applications has been very difficult. As a matter of fact the translations of proprietary BIMs from their own file formats to the correspondent IFC one, are not equivalent due to the different primary conceptual models of the building.

Even though different specialist actors use the same integrated application tool (e.g. Revit, Triforma, etc.), the entities they consider can have different meanings as belonging to different domains.

BIM has an important role in creating and coordinating components as parts of a building, but actually it does not provide any concept of the building as a "system" (structured set of components with functions aiming at a goal) like architects or engineers have had from centuries. Moreover other difficulties come from the fact that BIM data must co-exist with a number of programs with different task-oriented models, all essential in defining detailed information of a project.

IFC is based on a central model that can be either partially or entirely shared by participants, but must be accepted as a whole, being totally coherent (it is not scalable from this point of view). Although its approach supports different visualizations of the same concept, it is focused on converting and updating the integrated model from multiple sources, at the level of the applications, into a generalized description of the entire building.

As a matter of fact, current interoperability design problems related to commercial application programs are solved within the domain they have been built for, as very often they all have a similar, but particular, point of view: the one of who first modelled the phenomenon, probably some thirty years ago.

Models of a specialist domain allow "data exchange" but not "concepts understanding", so that the research work has been addressed to the specific issues that can enable a higher understanding level among actors.

Therefore the main problem this research has coped with, for improving collaboration in a cross-disciplinary design process, does not concern mere interoperability formats, but the mutual understanding itself of all the different actors over components, concepts and processes, which is still an open one: "the symmetry of ignorance".

3. Methodology

In such a scenario one important task of this research was to develop a methodology for a better mutual understanding by formalizing and managing knowledge in the broadest sense. This allows to integrate the fragmented specialist competences and expertise within an ongoing design project, reducing the "symmetry of ignorance" among the actors. The methodology used consists of analyzing the problem, defining a general model (a framework) of the design process, a common knowledge and a mechanism to interface common with specialist knowledge and to interface knowledge with data.

A fundamental pre-requirement to apply such a methodology and to overcome the above-mentioned mere interoperability format problems is *knowledge understanding*, namely technical knowledge. Technical knowledge concepts can be formalized and structured by means of the technology of *ontologies*, for defining entities; and by means of *explicit semantics* for defining their meanings.

Ontologies provide a valuable support for representing and sharing terminology, concepts and relationships within a given domain, so that an increasing number of communities [6] of experts develop ontologies as an underlying base for their work, including collaboration in design. Actually, beside the use of ontologies manually made explicit by actors, in the growing area of network services, new approaches to composition and orchestration of services are based on ontologies for representing their definitions and semantics, i.e. for disambiguating queries.

At present, most of specialist ontologies are typically not explicit and are inherent in the model of phenomena they are referred to, so that they are hidden in various application tools. i.e. in commercial tools, part of the knowledge is implicit, hidden-coded in application programs and is neither openly available nor understandable to users. It would be needed a reverse engineering…

Explicit semantics (context dependent meaning of knowledge) make entities understandable by humans and tractable by computers.

The only explicit semantics of most specialist application programs have, is at a lower level than IFC as it is linked to datum. Therefore, in many cases, only data can be shared in practice. As a consequence usual application programs that work on data at low semantic level, when applied to the complexity of the architectural product have generated more problems at a higher semantic level than those they had solved at the low one.

Although it was expected that an integrated application program could achieve interoperability among different domains of expertise, at present it is commonly recognized that the fragmentation of the design process increases the "symmetry of ignorance". Moreover it has become clear that a single data model, like integrated application programs, would not be able to serve all the requirements of all the actors and the sheer magnitude of the combined data often exceeds the capability of its management.

To deal with these problems actors have to be aware of entity meanings at the different levels of abstraction they use in a design process, from data level to reasoning level (Fig. 2). An ontology based methodology can also allows the actors to use in a coherent manner different levels of abstraction, or to exploit a conceptual interoperability.

4. Technology Description and Development

The assumption of our research is that actors cannot overcome the "symmetry of ignorance" barrier at data level or at low semantic level by means of usual application programs. Such a goal can only be achieved by means of two basic aspects: 1) matching concepts and mapping ontologies by rules to discover same entities in different domains [7] and 2) using inferential engines and intelligent agents to have an effective support in design.

Regarding the first aspects, at present, for machine interoperability purposes, translation of application programs to/from the common lower ontology level (IFC) is starting to be available - to the extent that software houses support new IFC specifications.

The dominant way of using IFC specifications (low-ontology level) today is still a one-direction batch translation of large data sets from an application into the common language and vice versa. Collaboration using IFC specifications exists in the industrial practice, but it is based on "ad-hoc" procedures that are agreed between single specialists at a project level. However, the nature of architectural and building design has proven to be a field of fragmented cross-disciplinary competencies, hard to manage, difficult to formalize in the above-mentioned utopian data-centric approach.

As a matter of fact IFC model servers implemented till now provide limited collaboration support and the existing model servers do not support adequate management of the instance versions (with different meanings) of various specialists.

To solve the aforementioned difficulties it is needed to postulate that each specialist has his own ontology (explicit or that can be explicited) and these ontologies have a partial overlapping meaning: a Common Ontology. Such an ontology can provide a base to capitalise the knowledge and expertise. In facts, matching current problems with past experiences requires to have stored them at the correct level of abstraction.

Moreover Common Ontology can be increased by the use of the semantic power of IFC. This can be done by adding new levels of intelligence: an Upper-Ontology Level (rules) and a Deductive–Reasoning Level [8] above the Ontology one (IFC) (Fig. 2). The first level allows parametric objects and logic/ algorithmic rules - that can adapt themselves to their design contexts - explain constraints, etc. The second level with its deductive capabilities applies inference rules and intelligent agents to the Upper-Ontology Level. This one can be triggered when ontologies at the Lower-Ontology Level match each other and are activated by instantiating a prototype of an ontology within the current context - and transitively, chaining rules as much as possible.

For example one actor may define the use of a room within a building and another actor may be warned when choosing a wrong size for a wall of that room or when a third actor proposes an incompatible material for the surfaces of that room.

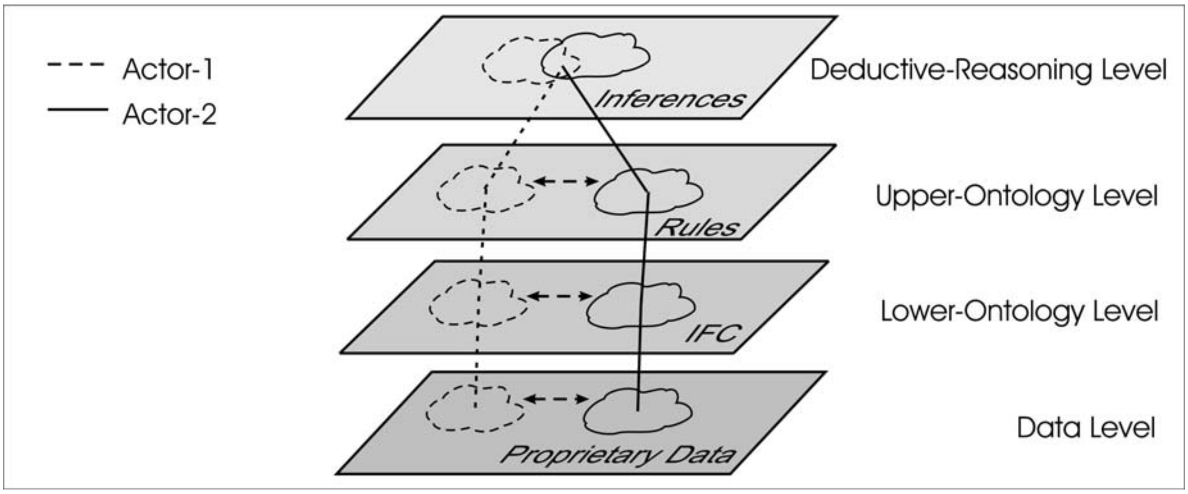

Figure 2 – Ontologies at Deductive-Reasoning Level map different Actor's ontologies at the Upper Ontology Level so that inference mechanisms can be applied to rules .

Concerning the second aspect, the use of inferential engines and intelligent agents, the basic differences of our approach with the present exhaustive and integral approach – consisting in importing and exporting in an agreed and identical format all the information, such as IFC (that anyway is a reference for the semantic matching) – consist of:

- exporting the strictly necessary information among actors;
- leaving to an upper-ontology based tool the task of linking representations of the same entities made by an actor at different levels of abstraction (vertical interfaces);
- making a Reasoning Level for mapping similar entities of different actors at the same upper-ontology level (orthogonal interfaces).

Referring to the first point, in contrast to a centralized database model a distributed ontology model has been developed, based over Personal and Common Design workspaces [9, 10, 11, 12]. Each actor's domain retains its own ontologies in the most appropriate form for his/her needs, while appropriate filters translate ontologies into/from a Common Ontology domain. The discipline-specific "ontology filter" translates the common representations into semantically-specific ones, as needed by their domains of expertise. Conversely, they will translate semantically-specific/domain-specific ontologies into a common ontology representation that can be accessed by means of other domain-specific filters. In turn, at the Low-Semantic Level data structures – representing specialist design instances by means of their own specialist "Data Filter" – are translated into common data structures (i.e. IFC instances) that make up sub-sets of the overall design instance.

To an actor, therefore, an ontology would have semantically specific representation, even though generated by a different actor, thereby facilitating a high level of shared common mutual understanding.

Regarding the second point, the research work has defined an ontology based model that supports actor's design by linking his/her heterogeneous semantics – whose formalization is oriented towards different tasks – with the data of his/her usual application programs, so that the model can point out inconsistency of data, incoherency of constraints, incongruence of goals within the actor's domain.

With respect to the third point, actors ahead of time can acknowledge the implications of their proposed actions, considering other actors' points of view, constraints and goals by means of inference mechanisms at Reasoning Level able to map constraint rules among the ontologies of different domains. The model will also let them achieve better and/or faster their own goals and global goals in decision making and problem solving activities.

The dynamic and semantically-specific representation can allow incoherent/favourable situations (according to different actors) to be highlighted and managed in real time. At the same time it allows actors to make alternatives, reflecting on consequences of their intents more consciously. So the impact of a networked ontology-based collaborative design transforms a hierarchical/linear fragmented process into a distributed and interleaved one, makes actors more aware of overall design problems and allows choices to be more participative and shared.

The integration of the specialist actor's design solutions (instances) translated into subsets of the overall design solution (instance) can give rise to inconsistencies and conflicts among instances belonging to ontologies of different domains. These inconsistencies are detected by the inferential mechanisms contained in the Deductive and Reasoning Level It first, points out (if it exists) a common ontology in the Common Ontology by mapping the ontologies of different actors, secondly, checks incoherencies among these ones; thirdly, reports feedback information to actors so that they can provide the necessary action.

The described Collaborative Working Environment has been tested by means of a real use case: a meta-design of a demonstrative Hospital Ward. The implementation is a demonstrative prototype system able to support a highly interactive collaborative design processes among three specialist actors in the field of Architecture, Structural and Mechanical Engineering [13].

Our framework uses a software system for the representation and querying of ontologies, named QuOnto (www.dis.uniroma1.it/~quonto/), developed in the past years at "Sapienza", University of Rome. This system, based on Description Logics, has proven to be computationally very efficient and robust enough to be used in productive environments (with million instances). We used Lisp for Reasoning Level rules, MySQL for the DB and two test bed threads for the development environment: SemanticWorks by Altova for ontologies + Architectural Desktop for graphic objects, or Protégé + SketchUp. See also Carrara e Fioravanti [9, 10].

5. Conclusions

In conclusions, it has been previously shown that all the above-mentioned difficulties can be overcome by means of:

- common and specialist ontologies;
- explicit semantics;
- an upper-ontology level for inference mechanisms;
- a disjoined overall coherence from actor's coherence;
- a storage of knowledge developed along the design process.

Making semantics explicit facilitates the management of design process at many levels. For example, the model can define a quality control level by means of key performance indicators (consistency, coherence, congruence of the design process), or it can manage intellectual property in collaborative design processes according to EU SA 8000 addressed on ethical impact of knowledge sharing.

The impact of the research is very large, as beside the developed use case of the Collaborative Design model in the field of AEC industry, the model has a non-specific structure to be applied to other specialist contents.

Thus conceived, the model confirms the possibility for any actor to work using his own personal methods, algorithms, software and tools to represent the complexity of his own instance of the design problem and to solve any contradictions in his own Personal Design Workspace. An actor has the added advantage that the other actors cannot enter his/her workspace although they are able to trigger yes/no these constraints/opportunities.

From the foregoing it emerges that the model structure is such that any actor defines the data of his own design instance by explicitly attributing them the conceptual ontology they refer to. In turn ontologies are structured by relations and rules in an upper-ontology level.

This evidently differs from what currently takes place in existing CAD systems in which the attribution of the data to concepts is implicit, arbitrary and related to the subjective interpretative capacity of the various actors.

Actually the direct link between ontologies (concepts) and data as well as the translation of both of them into ontologies common to all actors, make it possible to activate a true collaboration, in the broad sense of the term, on the basis of a mutual comprehension and the sharing of the choices.

References

[1] Wix, J.: 1997, ISO 10303 Part 106, BCCM (Building Construction Core Model) /T200 draft (1997).

[2] Gross, M.D., Yi-Luen Do, E., McCall, R., Citrin, W.V., Hamill, P., Warmack, A., Kuczun, K.S.:1998, Collaboration and coordination in architectural design: approaches to computer mediated team work, Automation in Construction, **7**(6), 465-473 (1998).

[3] Kvan T.: 2000, Collaborative design: what is it?, Martens, B. (guest ed.), Special Issue eCAADe ’97, Automation in Construction, **9**(4), 409-415 (2000).

[4] Jeng, T.-S., Eastman, C.M.: 1998, A database architecture for design collaboration, Automation in Construction, **7**(6), 475-483 (1998).

[5] Kolarevic, B., Schmitt, G., Hirschberg, U., Kurmann, D., Johnson, B.: 2000, An experiment in design collaboration, Automation in Construction, **9**(1), 73-81 (2000).

[6] Ugwu O.O., Anuba C.J. and Thorpe A: 2005, Ontological foundation for agent support in contructability assessment of steel structure – a case study, Automation in Construction, **14**(1) 99-114 (2005).

[7] Cheng M.-Y.: 2008, Cross-organization process integration in design-build team, Automation in Construction, **17**(2) 151-162 (2008).

[8] Nguyen T.-H, Oloufa A.A. and Nassar K.: 2005, Algorithms for automated deduction of topological information, **14**(1), 59-70 (2005).

[9] Fioravanti A., G. Carrara: 2007, Philosophy and structure of a CWE-based Model of Building Design – *in* P. Cunningham and Miriam Cunningham (eds), *eChallenges 2007*, ISBN 978-1-905824-05-2, IOS Press The Hague, Amsterdam, 24-26 October 2007, The Netherlands, 2007, CD Rom, pp. 1-12.

[10] Carrara G., Fioravanti A., Nanni U.: 2004, Knowledge Sharing, not MetaKnowledge. How to join a collaborative design Process and safely share one’s knowledge, *in* J. Pohl (ed.) *Intelligent Software System for the New Infrastructure*. Focus Symposium at InterSymp-2004 Conference Proceedings, Baden-Baden, Germany, 2004, vol. 1, pp. 105-118.

[11] Carrara G, Fioravanti A.: 2001, A Theoretical Model of Shared Distributed Knowledge Bases for Collaborative Architectural Design, *in* J. Gero e K. Hori eds. *Strategic Knowledge and Concept Formation*, ISBN/ISSN: 1-86487-114-8. conference Proceeding of the 3° International Workshop 17-18/12/2002, University of Sydney, Australia, 2001, pp. 129-143.

[12] van Leeuwen, J.P., van der Zee, A.: 2005, Distributed object models for collaboration in the construction industry, Automation in Construction, **14**(4) 491-499 (2005).

[13] Chen P.-H., Cui L., Wan C., Yang Q., Ting S.K., Tiong R,L.K.: 2004, Implementation of IFC-based web server for collaborative building design between architects and structural engineers, Automation in Construction, **14**(1) 115-128 (2004).