Yongwook Jeong, Armando Trento

Interleaving Semantics for Multi-Disciplinary Collaborative Design

Yongwook Jeong
Department of Architecture,
University of California at Berkeley,
232 Wurster Hall #1800,
Berkeley, CA 94720-1800, USA.
ywjeong@berkeley.edu

Armando Trento
a) Department of Architecture,
University of California at Berkeley,
232 Wurster Hall #1800,
Berkeley, CA 94720-1800, USA.
ywjeong@berkeley.edu

Yongwook Jeong, Armando Trento

Interleaving Semantics for Multi-Disciplinary Collaborative Design

Abstract

‘Collaboration is an important aspect of the architects’ education.’ (Kalay, Jeong 2001). The teaching of architectural design is facing with increasing urgency those aspects of the pedagogy related to the collaboration within the learning activity. The legacy of design as problem solving has been to consider collaboration a problem of effective communication where massive amounts of data must be shared among heterogeneous participants. Therefore, achieving interoperability among different CAD systems by way of organizing efficient databases has been the core research issue. The initial effort started with standardizing product descriptions including geometric information and constructing databases to organize them. The underlying theoretical assumption of these efforts is that *a building is a product composed of the heterogeneous products.*

This assumption has been relatively valid and even successfully realized in several related industries, such as the automotive and shipbuilding manufacturing industries. However, the building and construction industry continues to lag behind in this development because constructing a centralized product database has turned out to be not feasible because the shared database quickly becomes too large and unwieldy to support the dynamic process of multi-disciplinary collaborative design.

In contrast to the failed centralized database model, we propose to develop a distributed model, where each domain of expertise retains its own data in the form most appropriate for its needs, and where ‘intelligent’ filters translate data into and from a neutral data structure. The discipline-specific filters will transform the neutral representations into semantically-rich ones, as needed by their domains of expertise. Conversely, they will translate semantically-rich, domain-specific data into a neutral representation that can be accessed by other domain-specific filters. To the participants, therefore, the data they see will appear semantically rich, even when it was generated by another professional, thereby facilitating a high level of shared understanding. We also develop a computational methodology that will prove that computer-analyzed and generated design suggestions can actually help designers to achieve their goals better and/or faster. It will also let them know ahead of time what will be the implications of their proposed actions, as seen from other participants’ points of view. Once the participants contribute their knowledge to the representation, it would be possible that each of them could see the other’s point of view.

The dynamic and semantically-rich representation would allow incoherent/favorable situations to be highlighted and managed in real time and the participants to make alternatives reflecting their intents more effectively. The impact of a network-based collaborative design transforms a hierarchical/linear partitioned process into a distributed and interleaved one. In the filter medi-

ated communication model, the participating professionals can affect one another bi- or multi-directionally.

We propose the filter mediated communication model and its process to reflect the characteristics of multidisciplinary collaborative design without sacrificing human-centered aspects, and to solve real-world collaboration problems by focusing on a semantically rich representational method at three different levels that are mediated by intelligent filters. By fulfilling the discussed tasks, the experts from different disciplines participating in an AEC project are expected to better understand the dynamic process of design, to achieve a high level of shared understanding. and to facilitate the onset and dissemination of creative ideas.

Introduction

The legacy of design as problem solving has been to consider collaboration a problem of effective communication where massive amounts of data must be shared among heterogeneous participants. Therefore, achieving interoperability among different CAD systems by way of organizing efficient databases has been the core research issue. The initial effort started with standardizing product descriptions including geometric information and constructing databases to organize them. Following the standardization of product model data, Eastman (1991) proposed Engineering Data Model (EDM) to manage heterogeneous information carried by different design and engineering applications.

The underlying theoretical assumption of these efforts is that *a building is a product composed of the heterogeneous products.* This assumption has been relatively valid and even successfully realized in several related industries, such as the automotive and shipbuilding manufacturing industries. However, the building and construction industry continues to lag behind in this development (Tolman 1999). In the following sections, we will examine the Building Product Model (e.g., ISO-STEP, IAI-IFC) and its problems, and discuss the Filter Mediated Communication Model as an alternative approach.

Building Information Modeling

Building Product Models

In the 1970s, the National Institute of Standards and Technology (NIST) proposed a standard for the exchange of massive geometric data, called the Initial Graphics Exchange Standard (IGES), which defined a neutral data format as the lowest common denominator among CAD systems that use it. While IGES has been developed and maintained by a governmental agency, Data eXchange Format (DXF) has been developed by Autodesk in order to meet customers' needs. The technical objective of this approach is to transfer one CAD system's data to another and vice versa through the least amount of data extracted from available CAD systems and other engineering applications. However, as CAD systems grew more diverse and sophisticated, this approach soon revealed its limitation. The most notorious problem is that when one application's data is translated into one of the neutral file formats, the translated data is no longer consistent with the original data because it loses the semantic information which was relevant to the application.

The complexity of the file formats is another problem. Both IGES and DXF require solid professional programming skills to understand and manage. These weaknesses were not easily corrected so that searching for a new method was accelerated. With this motivation, the International Standards Organization (ISO) developed an international standard in 1984, known as the Standard for the Exchange of Product Model Data (STEP) for computer-based description and exchange of physical and functional characteristics of products throughout their life cycle, independent of any particular system. A product model is an information model that implicitly contains data regarding form and function of a product and aims at describing the targeted product through its life cycle. A building product model is an example of a product model in the AEC industry, which can describe the form (e.g., geometric information and its relationships) and function (e.g., energy performance) of a building through its life cycle. Recently, these efforts were reincarnated as Building Information Modeling (BIM) driven by several CAD system vendors (Autodesk, GraphiSoft, Bentley, etc.).

Industry Foundation Classes

The newest and largest effort is collectively known as the Industry Foundation Classes (IFCs), published by the International Alliance for Interoperability (IAI) in 1995.

The IFC is an open and non-proprietary data model specification in the AEC industry representing a fixed set of objects commonly used for the built environment. IFCs are used by computer applications (not intended for humans) to assemble a computer processable model of the facility that contains all the information of the parts and their relationships to be shared among project participants.

The philosophy of the IFCs is to electronically represent all possible aspects of a building including products (e.g., doors, walls, fans, etc.) and abstract concepts (e.g., space, organization, process etc.). These specifications represent a data structure supporting an electronic project model useful in sharing data across applications. Each specification is called a ‘class.’ The word ‘class’ is used to describe a range of things that have common characteristics. For instance, every door has the characteristic of an opening to allow entry to a space; every window has the characteristic of transparency so that it can be seen through. Door and window are names of classes.

The classes defined in the IFCs model can be used in designing a structure, in costing and scheduling, in providing critical data appropriate for contractors and subcontractors, in evaluating energy performance, or allowing facility managers and building owners to access data pertinent to their business.

Problems and Limitations of BIM

For the past few decades, it appeared that such product models could serve the purpose of collaborative design. Eastman (1997, 1998) proposed a universal building model, which was an attempt to support multiple designers using different applications forming an effective multi-user collaborative design environment. It is based on a central model that can be either partially or entirely shared by participants, and specific design ‘views’ which are defined as units of organization specific to each participant.

Although his approach supports different views, it is focused on converting and updating the integrated model from multiple sources at the level of the applications themselves into a generalized description of the entire building. Eastman's, and other centralized data models, were well defined and tractable enough to support a limited number of participants. However, the complexity of the architectural product has generated more problems than the shared data model could solve. The IAI have also identified the most time consuming aspects of their IFC definition problems as follows (Wix et al.):

- Ensuring as wide an agreement as possible within the industry on semantic definitions.
- Obtaining model reviews, handling the issues that result from review and ensuring that issues resolution is open for all members to see.
- Integrating all domain developments into a single model that is internally self consistent.
- Ensuring technical consistency across the whole of the model and all of its supporting documentation.
- Providing guidance to implementers to ensure minimum ambiguity within the model.
- Developing the necessary testing mechanisms to ensure that there is a means of guaranteeing compliance of software with the objectives of IFC design.

The listed difficulties would be true of all the integrated model approaches. It has become clear that a single data model would not be able to serve all the requirements of all the participants. In addition, the sheer magnitude of the combined data often exceeds the capability of its management by any one domain. Although an integrated model was expected to achieve interoperability among different domains of expertise, it actually exacerbated their fragmentation and the symmetry of ignorance.

The nature of architectural design has proven to be an obstacle to the kind of utopianism embodied in the data-centric approach. It can be likened to the making of a puzzle, where designers search for individual solutions (spatial, structural, material, economical, etc.) that can fit together in some spatio-temporal context. It is an iterative, dynamic process, where propositions are made and tested against goals and constraints that are both internal to each domain of expertise, as well external to them (i.e., originating in other domains).

Filter Mediated Communication Model

Rationale

When the participants in a design process make decisions and negotiate with one another, they use their own representations, knowledge, methods, and resources. Take three participants in the design of an office building: an architect, a structural engineer, and a mechanical engineer. The architect designs an initial layout and one or more lists of requirements and constraints, using his own representational methods. The architect "publishes" the drawings in a way that can be "read" by the structural engineer. Based on the architect's input, the structural engineer designs the structure of the building using his own knowledge and method of representation. The structural or mechanical engineer most likely will review the architect's drawings first, and design the structure or HVAC using structural or energy codes and standards as well as his own disciplinary tool. Once his design is ready, the engineer “publishes” his design. However, if he is unable to design within the range of the architectural drawings, the participant will ask the others for modifications. This scenario is, of course, only a brief description of a rather lengthy and iterative process, highlighting only three of the many participants. It can be easily complicated by adding more participants. However, it sufficiently shows how each participant in the process alternates between their ‘private’ representations, used during their own, internal design process, and the ‘public’ version which they ‘publish’ for the benefit and use of the other participants.

The core of collaboration arises when all the actors can see the different views within the structure of the product’s model as a whole, analyze their inconsistencies, understand each others’ reasons and see their effects on the whole as well as on their view, evaluate others’ proposals, assess the possible trade-offs, and eventually modify their own view.

This kind of process has to be directly interactive, in order to allow what can be called a “mute discussion”. Two requirements are therefore requested: a complex representation of the shared model, structured by views and including objects and mutual relationships; a means to mediate a confrontation among different (or concurrent) ideas of the different actors. The representation of the product’s model as a whole in the SDW is thus a complex one, as much structured as possible, and in the meantime must be poor enough (“lean” model) to be understandable by all. To allow the process to be truly collaborative, modifications should be targeted and real-time interactive, mainly in the ”creative” moments; any actors may well not know why another one has done a modification but they certainly have to understand the effects, mainly for what they are concerned.

We assume a kind of ‘filtering’ mechanism mediates between the private (domain-specific) workspace and the shared public workspace. This filter strips the published version from each of the participants’ private notations, sketches, calculations, and other design representations that the participant uses during the design process, and which would be of no use to the other participants. A similar but inverse filtering occurs when each participant receives the input generated by the others, and interprets and translates it into their domain-specific representation.

We contend that it is this ‘filter’ where much of the essence of collaborative design resides. It is the process which adds disciplinary knowledge to input received from other participants, and which makes each participant’s own discipline-specific representations easier to comprehend by the others (much like a perspective drawing is a form of representation that makes it easier for the client to understand the architect’s design). The ‘filter’ is thus a bridge between the private workspace of each participant, and the public, shared workspace. In the public space, the object level of information exists, which includes geometric and non-geometrical data as well as semantic data attached to them. In the private space, there exist two levels: the conceptual and the mechanical level. The participant’s tacit knowledge and beliefs are in the conceptual level. In the mechanical level, ontology (a formal specification in the conceptual level), value systems (the subjective and disciplinary value in which each expert views the performance of the product), and various tools exist. The filter mechanism operates on top of these two levels.

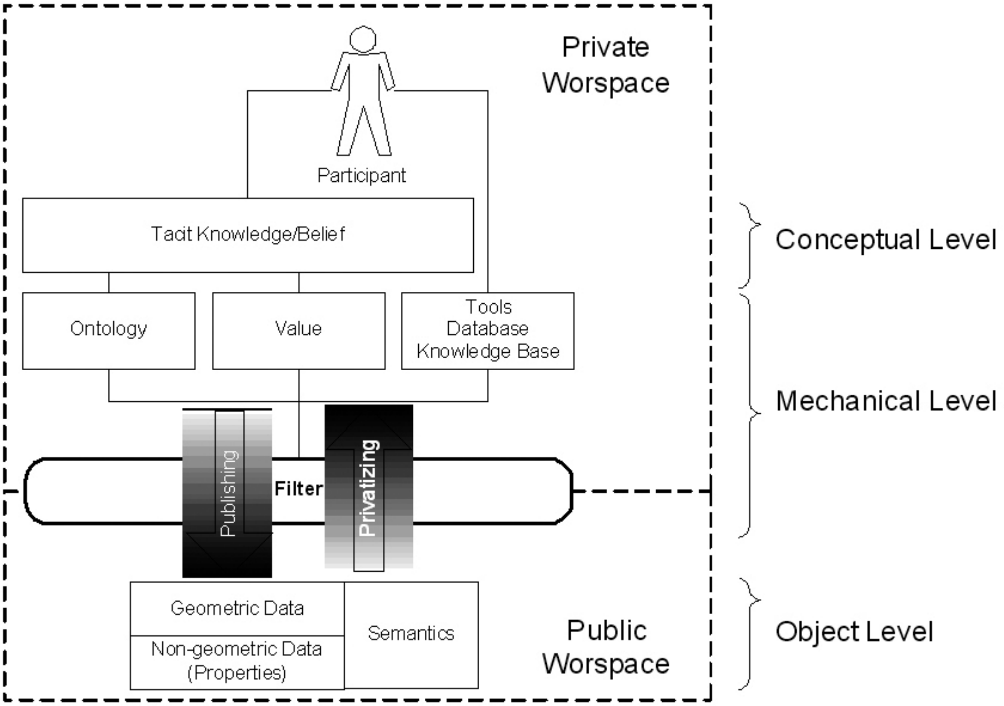


Figure 1

Anatomy of a discipline

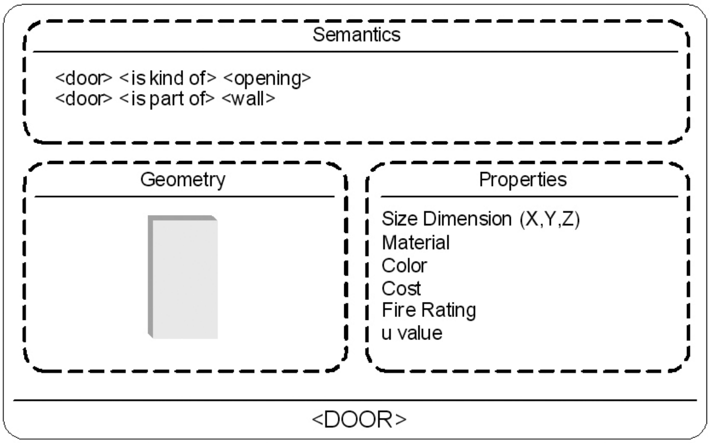


Figure 2

An ontology of “DOOR”

This approach matches conventional multi-disciplinary design processes. In conventional design processes, such ‘filters’ are the knowledge used by each of the participating professionals. It differs from the data-centric models of collaboration in that it uses no centralized database. Rather, the project data is a collection of the individual contributions of the participants. If any one of the participants wishes to see data produced by another participant, they have to retrieve the relevant information from that participant then translate it into its own representational form within its own workspace. The entire project can thus be regarded as a compilation of each individual’s own representations. It comes from the *interleaving*, of the ‘filtered’ private views linked by their mutual relationships.

Each one of the participants is responsible for authoring their own representations, and for retrieving the latest version of the design produced by the other participants. At any stage, but at the very end of the design process, each contribution can be incoherent, thus the solutions displayed in the shared workspace can be inconsistent both inside and among themselves. The role of the ‘filter’ is to facilitate the participants work, *mediating* their different representational forms and supporting the *integration* of their individual contribution on the shared product.

Ontology as an Extended Representation

The filter operates as an autonomous entity – an agent – that can take on partial responsibility for exchanging information with other computational or human experts who have domain-specific knowledge. To do so, the filter uses its own built-in domain-specific knowledge. That knowledge is represented as an ontology – a concept borrowed from philosophy and given a specific technical meaning in computer science. An ontology is a formal, explicit specification of a shared conceptualization: a common vocabulary and model of some concept or entity (Gruber 1993). An ontological framework can be used by a set of actors (humans, software applications, artificial agents) so that they can communicate about the domain of discourse without necessarily using a globally shared theory (Katranuschkov, 2003). In the context of the AEC industry, an ontology can define a set of representational terms in some knowledge domains (architecture, structural engineering, mechanical/ electrical/plumbing, construction, etc.).

These definitions associate the entities of the domain of discourse (walls, doors, columns, beams, joints, bearing structures, and the like) with human-readable text describing what these entities mean, and formal axioms that guide and constrain the interpretation and the use of the terms. Thus, an ontology adds to the descriptive data logical statements about the nature of the described entity itself. For example, if a data model describes a door in terms of its geometry, material, cost, and other descriptive parameters, an ontology of doors adds concepts like controlled passage, lockability, and the (necessary) relationship between a door and the wall in which it is embedded. This expanded description provides a common vocabulary that defines meaningful queries and assertions about the represented entity (e.g., does the door retain its ‘doorness’ when it is disassembled from the wall in which it is embedded, and used as a table top?).

To be more explicit, let us consider what may happen when information about a door is needed, for example, to check its code compliance for fire egress. In a data-centric (e.g. IFC-based) environment, this information would be stored in a shared project repository as a DOOR object and a number of related resource objects, such as MATERIAL, PROPERTIES, SHAPE, etc. An appropriate query can easily be formulated, even answered. However, the answer might be meaningless, if the door is not located in a wall, or it is locked or blocked and cannot afford egress. In contrast, the ontology approach would allow us to define the concept of a door fully, in a manner that does not depend on the 'intelligence' of applications that use it. It would provide the sufficient means to execute operations on door objects, and interpret the results of these operations correctly. Furthermore, it would be possible to define the 'door' object itself for a specific context. Any geometric object with specific properties including performance can be regarded as a door (Figure 2).

The door ontology describes not only geometric/non-geometric information, but also the semantics, which enables to retain the underlying ideas. According to the semantics of the door ontology in Figure 2, a door can retain its doorness only if it is embedded in the specific wall.

To fulfill its desired function as a means of high-level communication among the participants in an AEC collaborative design project, the filter must behave in an intelligent manner. It must operate both at a syntactical level (i.e., allow different applications to read the data), as well as on a semantic level (i.e., convey the ontological meanings associated with the data). More specifically, we have identified two different levels of communication in the object level:

- Semantic level – expressing ontologies and relations (internal or external mappings) for a particular domain.
- Syntax level – machine-processable format, such as XML.

Semantic Level

The semantic level is the first step towards inter-domain communication. It adds to the object data the conceptual ontology that any one domain expert may take for granted, but which would be viewed differently by another domain expert. It thus provides a more explicit, but abstract way to describe information, encapsulating both conceptual and domain-specific data models. The conceptual models may include elements such as generalization, aggregation, and cardinality constraints about the objects (e.g., that a door is a kind of opening and belongs to a specific wall). The domain models deal with vocabularies defined by domain-specific ontologies, such as architecture, structural engineering, mechanical engineering, and general contractor to name a few.

This level fulfills the most important role in the filter model: adding semantically-rich information that can be interpreted correctly by different domains of knowledge. As an example, consider the design of a house. The plans produced by the architect include objects such as walls, rooms and openings. The structural engineer must interpret the plans, retrieving the meanings of the objects it contains. However, the structural engineer's interpretation is often different from the architect's: He uses different vocabularies, such as "bearing walls", "partitions", and "frames." The structural engineer may, therefore, begin his task by translating the objects from the architect's representation into his own objects, creating a totally different representation of the same floor plan. The architect will have to go through a similar process when he receives the structural engineer's drawings.

The proposed filter mechanism will automate this interpretive/translation process, using the semantics associated with each object that comprises the plans. It will interpret, add, or omit data as needed by the domain expert. For example, the structural engineer's filter will interpret architectural WALL objects as LOAD_BEARING or NON_LOAD_BEARING objects, without burdening the architect's representation of the same data. With the addition of the suggestion-based applications, which will be discussed later, the architect may be alerted when he tries to modify a wall designated LOAD_BEARING by the structural engineer, but the data describing the wall's specific load bearing properties will be hidden from him.

This process requires that some or the whole of participants share a level of ontology to be able to discuss about the same objects or concepts, as for instant the WALL of the previous example: this entities have to exist in more than one domain and they have to share part of their definition.

Syntax level

This level is intended to provide a common standard for exchanging data. We propose to adopt eXtensible Modeling Language (XML) as a common syntax. The main task, therefore, would be to tag each object using appropriate XML tagging. The tags alone will not carry meaning unless they are connected to ontological information that is stored in the semantic level. Given a common syntax, given the participants share a level of project-independent ontology, the access to multiple views is possible only if each of the participants defines explicitly both his internal and the external connections to the ontological information. The filter mechanism will support this design operations facilitating the participants in building their own semantic structure and "subscribing" some relations to the others.

Communication Modes

In terms of computer networks, there are two common models of computer networking: the client-server model and the peer-to-peer model (Barkai 2002). In a client-server model, the client (the user's computer) makes requests of the server to which it is networked. The server, typically an unattended system in a back room, responds to and acts on the requests. Data-centric approaches usually use this model. On the other hand, the idea behind the peer-to-peer model is that each 'peer,' i.e., each participating computer, can act both as a client and as a server.

The notion of decentralization is directly applicable to this model. Figure 3 shows the filter mediated communication model with published data contributed by each participant. Pure peer-to-peer computing has no central server and router, and peers in this network act as clients and servers. Hybrid peer-to-peer has a central server that keeps information on peers and responds to requests for that information. Peers are responsible for hosting the information, letting the central server know what information they want to share and for downloading its shareable resources to peers that request it.

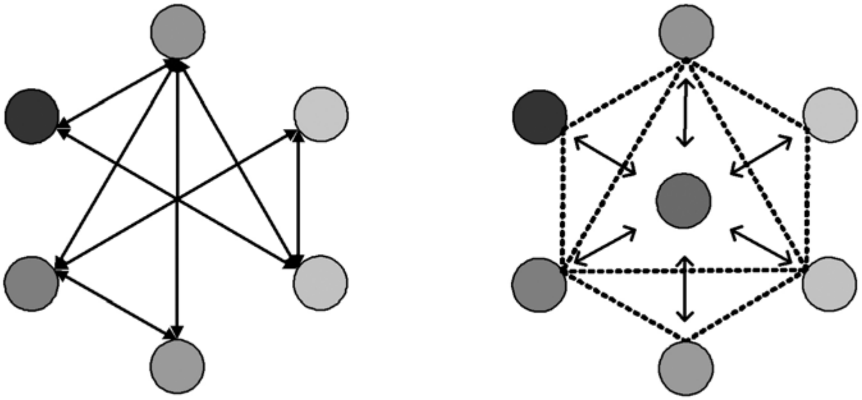


Figure 3
Two communication modes
pure peer-to-peer (a) and hybrid peer-to-peer (b)

The major advantage of peer-to-peer models over the client-server model is that we can reduce the size and complexity of the centralized data and even eliminate centralized control over the data, which is expected to overcome the problems of the data-centric approach discussed earlier. For practical purposes, we adopt the hybrid peer-to-peer model for filter communication. This is because in the pure peer-to-peer model, peers may spend much time and effort to find other peers and their resources. The shared data includes general information on peers, agreements on how to describe ontological specifications for the specific domains, and the latest version of each participant's published data.

We regard each participant's filter as a peer in the filter communication. When the filter obtains a connection established between the server which has the shared data, it examines the consistency of its maintained published data. First, it will publish the latest detected version of each participant's data. Second, if it detects inconsistency between the published data and the private data, it will update its published data. Since each participant's private data is dynamically changed in the course of the design, the filter will update its own published data when it is requested to provide information by other filters.

When a participant's filter connects to the server in order to retrieve the published data created by other participants, it will immediately establish a special relationship called "subscription." This will facilitate communication among filters. If the filter subscribes other published data, it will constantly check all new information on the published data so as to notify the participant and respond to the changes in an appropriate way. Although there are some changes to the published data in which the filter is interested, they may be able to be ignored if they do not affect the project significantly. For this purpose, the filter has a mechanism to specify interest criteria to determine whether the changes are relevant or not.

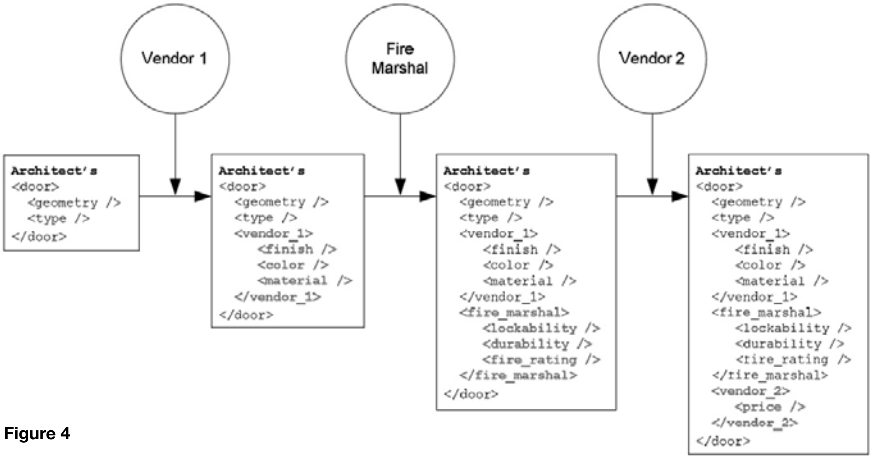


Figure 4

Participant-oriented representation of an object

For example, if the architect has subscribed the structural engineer's design and he is interested in the diameter of columns, he would specify an interest criterion (e.g., if the diameter exceeds 10, then notify me). Based on the criterion, the architect's filter will not notify until it violates the dependency.

This distributed communication model envisions the filter mediated communication model in two ways. First, it is an implementation of the notion that each individual's own representations comprise the entire project and they are responsible for producing their own representations. Second, when each individual's filter receives others' published data which it does not understand, it asks others' filters for ontological information connected to the published data.

Discussion

Participant-Oriented Representation

Every participant creates their own objects by specifying geometric and non-geometric information as well as ontological information (or semantics). Consider that the architect designs a building with his own objects. In the schematic design phase, his objects would be generic in that they only describe the function or performance and some of the simple geometry specific to that object. It would be used to define spaces, enclosures and openings while he develops some architectural plans of the building. In the course of subsequent phases, more detailed specifications could be determined and added to the objects. For example, a door in the preliminary design phase would be used to define an entering point to analyze circulation. At this point, no specifications are needed. The color, material, finish, sound, fire-rating, and price might not be taken into account. In the design development, the architect publishes the plans and the same door object would be specified in more detail by the participating vendors. Figure 4 shows a participant-oriented representation of the architect's door incrementally enriched by the pieces of the other participant's descriptive knowledge. As shown in the case study, a structural member would be treated in the same way. In the schematic design stage, it would be modeled as a generic object. With time, the member would be more articulated and properly dimensioned with the help of the structural engineer. Once the participants contribute their knowledge to the representation, it would be possible that each of them could see the other's point of view. The dynamic and semantically-rich representation would allow the participants to make alternatives reflecting their intents more effectively, which eventually leads to a state of shared understanding.

Distributed and Interleaved Communication

As discussed earlier, the impact of a network-based collaborative design transforms a hierarchical/linear partitioned process into a distributed and interleaved one. where the sequence of inputs is not pre-determined, but rather opportunistic. In the filter mediated communication model, the participating professionals can affect one another bi- or multi-directionally. It means that opportunities can be recognized and acted upon in time to make the most of them, and problems can be spotted earlier, when they arise, because more specialists will have access to the evolving product: they will not have to wait their turn to be consulted, at which time it may be too late to recognize an opportunity or to avoid a problem.

Figure 5 shows that each participant has their own model which is relevant to each stage of the project, where the continuous line indicates the bidirectional relationships mediated by the filters and the dashed line illustrates their contribution to the design process. The continuous interaction among the actors and/or their filters, allows incoherent/favorable situations to be highlighted and managed in real time, and may facilitate the onset and dissemination of creative ideas.

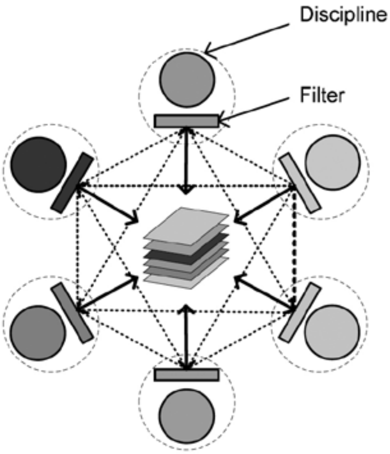
The participants do not have to share a large and heavy integrated model. Rather, their intelligent filter will access the information in the object level (geometric/non-geometric information and ontologies) which resides at its own location and translate it into their own representation using user-defined ontologies.

They would fill the gap between the heterogeneous representations preserving semantics as long as they are based on the syntactical agreement (e.g., XML). While producing and consuming the information, the participants and their filters construct a "knowledge chain" in which subassemblies of the information are passed from one filter to another, each one contributing its own piece of knowledge.

We propose the filter mediated communication model and its process to reflect the characteristics of multidisciplinary collaborative design without sacrificing human-centered aspects, and to solve real-world collaboration problems by focusing on a semantically rich representational method at three different levels which are mediated by intelligent filters. By fulfilling the discussed tasks, the designers from different disciplines participating in an AEC project are expected to better understand the dynamic process of design, and achieve a high level of shared understanding.

Figure 5

The filter mediated communication model



Reference

Barkai, D. (2002). Peer-to-peer computing : technologies for sharing and collaborating on the net. Hillsboro, OR: Intel Press.

Carrara G. and Fioravanti A. (2002). 'Private Space' and 'Shared Space' Dialectics in Collaborative Architectural Design, Collaborative Decision-Support Systems, Focus Symposium and InterSymp-2002 Conference Proceedings, Baden-Baden, pp. 27-44.

Carrara, G, and Kalay, Y.E. (1994). Past, present, future: process and Knowledge in Architectural Design, in G Carrara and YE Kalay (eds.), Knowledge-Based Computer-Aided Architectural Design, Elsevier Science Publishers B.V., Amsterdam, pp. v-vii, 147-201, and 389-396.

Eastman, C., Jeng, T. S., Chowdbury, R., and Jacobsen, K. (1997). Integration of Design Applications with Building Models. Paper presented at the CAAD Futures '97.

Eastman, C. M., Bond, A., and Chase, S. (1991). A Data Model for Engineering Design Databases. Paper presented at the 1st International Conference on A.I. in Design, Edinburgh, Scotland.

Gruber, T. R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing: Knowledge System Laboratory, Stanford University, CA, USA.

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

Kalay, Y. E. (2004). Architecture's new media : principles, theories, and methods of computer-aided design. Cambridge, Mass.: MIT Press.

Katranuschkov, P., Gehre, A., and Scherer, R. J. (2003). An ontology framework to access IFC model data. ITcon, 8(Special Issue eWork and eBusiness), 413-437.

Korman, T. M., Fischer, M. A., and Tatum, C. B. (2003). Knowledge and Reasoning for MEP Coordination. Journal of Construction Engineering and Management, 129(6), 627-634.

March, J. G., Simon, H. A., and Guetzkow, H. S. (1958). Organizations. New York,: Wiley.

Mohsini, R. A. (1991). On Measuring Project Performance: Some Problems of Aggregation. In Evaluating and Predicting Design Performance. In Y. E. Kalay (Ed.), Evaluating and Predicting Design Performance (pp. 239-249). New York: Wiley.

Tolman, F. P. (1999). Product modeling standards for the building and construction industry: past, present and future. Automation in Construction, 8(3), 227-235.