The Quest for the Holy Grail – Holistic Collaborative Design

How to enhance communications among tools of semantic different level in a cross-culture design.

Gianfranco Carrara; Antonio Fioravanti
Dip. Architettura e Urbanistica per l'Ingegneria
University of Rome "La Sapienza"
http:www.dau.uniroma1.it

Abstract. *Architectural design, due to phenomena such as globalisation of the construction process, delocalisation of professional and industrial activities, spread of new construction materials and components, and the challenges of environmental sustainability, has become so comlex that traditional ways of managing the process are no longer sufficient. What is more, architectural works are required to be ever more performing and integrated - holistic. The paradigm of collaborative design is gaining ground as a way of dealing with these problems. Various process/product models have been proposed over the years, making more or less use of advanced tools.*
In this field this study proposes a model in which operators from various cultural contexts are each supported by Intelligent Assistants (agents). The model simulates the design of works of architecture to the best professional practice. Exchange of information and knowledge between the operators is essential to enhance the design process. From this follows the need to interface not just data, but also meanings.
This article shows how to improve communications between different applications, used by different operators, so as to integrate information and knowledge, whether formalised or not, in a project managed collaboratively by means of XML.

Keywords. *Collaborative design; Design Process Model; distributed knowledge bases; semantic interfaces; aecXML.*

Architectural design and communications problems

In our times, architectural design is a collective process formed by the intervention and interaction of numerous professionals, as well as indirectly by the action of non professionals. Architectural design is part of the broader construction process in building which leads to the construction, restructuring and ultimately the demolition of building assets. The overall process is fragmentary owing to the different "actors" [1], different phases and regulatory, environmental and cultural contexts. All this introduces an inefficiency of costs and/or timing, or leads to buildings that do not attain the required performances.

Very often cost overshoots of up to even 100% occur and it is not unusual for the length of time required to realize and accomplish the work to exceed initial estimates by as much as 50% [2]. Furthermore, a decline is often observed in the final project quality due to the impoverishment and/or deviation of the initial idea in respect to the actual realization, caused by the adoption of simplistic solutions in the preliminary project. This is owing to a misunderstanding of the technical-constructive-plant engineering problems (or opportunities[3]) which are viewed as separate stages incorporated in a serial process. Indeed, due to limitations of time, alternative designs are often not proposed nor evaluated during the preliminary design phase, with the result that in the detailed and constructive phases, the design can turn out to be inadequate.

The above-mentioned drawbacks may lead to: a lack of coordination of the activities performed, different design objectives for the various actors, and difficulties involved in communicating the reasons underlying design choices of the actor (Borkowski et al., 2001, Garner and Mann, 2003).

These difficulties are largely due to problems involved in communicating and understanding different models of reality used by the actors to highlight the aspects they are interested in (Woo et al. 2001). Clearly it is not only a problem of amount of information exchanged but, above all, of its interpretation and the associated meanings (Kalay, 2001; Garner, 2003).

ICT models for bridging the information/knowledge communications gap

To reduce the highlighted design drawbacks, companies have attempted, by gradually increasing the number of professionals and specialists working in their various branches, to increase the skills available and to short cut the decision-making processes by means of meetings among the actors: collaborative design. It is apparent how, as the number of employees in a design office increases or when interaction with other design offices is essential, the above-cited problems recur.

A second approach to reducing the cited problems, which is parallel rather than alternative to that considered above, is the development and use of ICT support tools.

This path, followed by several software houses (Williams, 2003), or service companies (Laiserin, 2000; Laiserin, 2001), consists of managing all the information by oversizing the existing applications software (Khemlani, 2003), which ultimately led to the concept of the super-database (super, in the quantitative sense) and is unique, totalizing and rigid (Pohl and Myers, 1994).

The difficulty of finding, at the higher semantic level, tools to support communications between

[1] *Note that an 'actor' is "Any participant in the design process", and it is used in this sense in this paper (Wix, 1997).*

[2] *For example, the new Centro Congressi all'EUR by M. Fuksas at Rome for the detailed design phase, and the Auditorium by R. Piano at Rome for the constructive design phase.*

[3] *Problems, due to the specific requirements of an actor arising from his professional experience or from the introduction of new technologies, often reveal themselves to be suggestion for new design statements, which when are accepted, result in new opportunities for the design of work of architecture.*

the applications and hence the actors themselves, lies in the fact that any enhancement of the knowledge present in the project corresponds to a parallel further complication of the corresponding DB and the need to formalize all the design processes and construction components (Pohl, 2004; Pohl, 2002; Kim et al., 1997)

We propose an overall different model for Collaborative Design, the underlying philosophy of which is alternative to the preceding models. In this model not only data and prescriptions are exchanged but above all ways and means of extracting the shared design objectives from the problems/opportunities phase by phase (Carrara et al., 2004; Carrara e Fioravanti, 2004).

This is a huge challenge, which well merits the title of "Quest for the Holy Grail". And in the development of increasingly powerful CAD tools, having a system capable of effectively and concretely enhancing architectural design, has always been an objective.

The model which we have been developing for many years to achieve this objective is starting to emerge (Carrara et al., 2001; Carrara and Fioravanti, 2001). In this model the actors are helped by relatively 'intelligent assistants', I.A., capable of providing support in relatively complex tasks.

This paper is a contribution to the definition of another tessera in such a mosaic: the communication among actors from different cultures – which is much more than just an interface among applications of heterogeneous environments [4].

The model allows both formalized and non formalized information to be transmitted and takes into account the different points of view of the actors, as well as following a holistic approach in dealing with the problems/opportunities; it is flexible enough, in the course of the project, to be able to take into account new characteristics and/or constraints imposed by ICT objects representing the building components and the design process (Cheng, 2003).

The model proposed is a 'mixed' model, since not all aspects can be formalised [5]. For this reason, formal notifications have been associated with a "whiteboard" to communicate intentions and how information has been structured by means of XDT (XML Data Type).

For the first set, it is relatively simple to automate the exchange of data and information using ICT tools, since their ontologies and process rules have long been defined, as actors act in specialist domains; the result has been to facilitate the formalisation of 'specialist knowledge'. The second set is outnumbered by 'commonsense rules', hence of 'general knowledge domains', so that it is much more difficult to realize effective formalisations on the basis of knowledge alone [6].

One of the difficulties in achieving such an ambitious objective is the quantity of data required if our information and knowledge are to be significant.

The novelty of the model consists in the fact that, although it handles semantically rich ICT objects, the latter are not resident in a super-DB, but are virtualized in several applications, KBs and DBs.

The objects representing the truly shared (or shareable) part consist of only a small core of knowledge compared with the whole.

[4] In this regard, a new approach called Aspect Oriented Programming, AOP, is currently emerging, although it is still too early to provide us with tools suited to our needs.

[5] As well known from Gödel's theroem (Hofstadter, 1988).

[6] It has been shown that "sentient" tools are required in such cases (Tagliasco e Manzotti, 2003) in order to resolve ambiguities (Simon, 1997, pp. 77-80).

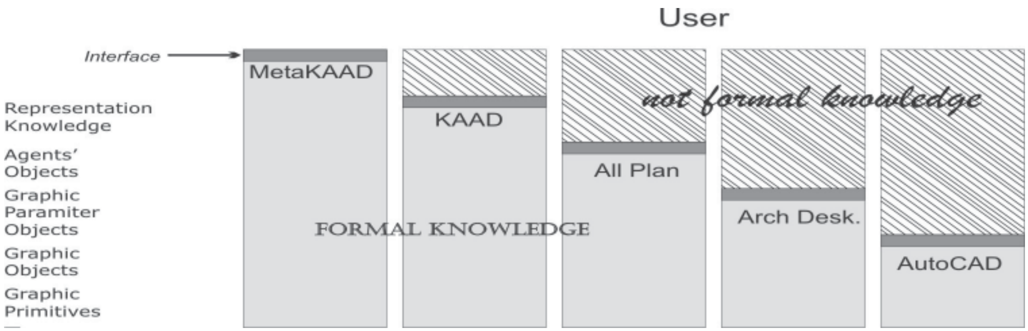

Figure 1

'Intelligent level' layers of architectural ICT tools

The stratification of "levels of intelligence" of the tools used is conceptually similar to that of the ISO-OSI stacks for ICT communication protocols.

Indeed, in the same way, in the assisted design field, one goes from the level of Intelligent Assistant, IA (formed by Metaknowledge, Knowledge Base, inference engine, UI, DB), and specific applications software to parametric modelers and so on down to the lower level, that of graphics primitives (Fig. 1).

For the above reasons, we introduce the concept of 'variable intelligent level stratification' interface of the IA between the MetaKAAD environment with which the system dialogues with its human users and the applications used by them, which enables communication among different tools with different 'levels of intelligence [7]'.

The model specifies a variety of Filters/Perspectives between the different levels, for example, AutoCAD <-> KAAD [8]; or between equipollent levels with low semantics, ARX objects <-> Shapes of the KAAD agent objects; or between equipollent levels with a high semantic level, aecXML agent objects <-> KAAD.

The mechanism of interoperability among the various "levels of intelligence" of the models used to represent reality, with which the applications of the various actors comply, render unitary the overall Model deriving from the integration of all the partial models.

Interoperability is not an end in itself but is needed in order to transmit data, meanings and knowledge.

This is done in three ways: syntactic analysis, formalized knowledge, non formalized knowledge.

The three ways correspond to three tools: the PARSER (composed by a lexical analyzer, Flex and a syntax analyzer, Bison), the XML (for standard management, standard and non-standard information), and the Whiteboard.

The interoperability of the model to enhance holistic collaborative design

Analysis of interoperability problems

This study, along with the problem of communications, faces up to the problem of interoperability in architectural design. To this end, we have developed a prototype system, called MetaKaad, which integrates the contributions of a number of

[7] *In a rough definition, intelligence = semantics + rules + MetaRules.*

[8] *KAAD is a knowledge based system developed by us at CAADLab – Dpt. Arch. and Urban Plann. – University of Rome "La Sapienza" (Carrara and Kalay, 1993; Carrara et al., 1995).*

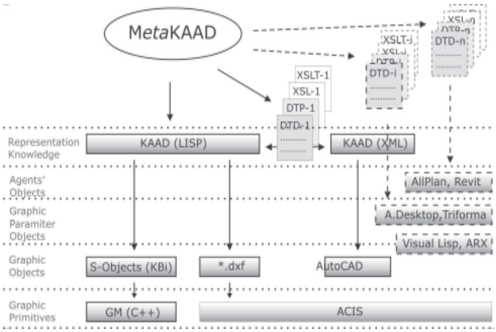

Figure 2. An example of an interface between MetaKAAD and AutoCAD with 'variable stratification'.

designers collaborating on a single project. We asume that they continue to use the techniques, methods and software tools they have learned in the course of their work and which are customary in their professional context. MetaKAAD allows software tools with diverse record structures, working towards a convergent design objective, to exchange data about the project in an effective and consistent manner.

We show below a schematic collaborative design scenario which, for graphic and simplicity reasons, includes only two actors (an Architect and a Civil Engineer) although it can be generalised to a larger group, and indicates the software architecture supporting their interaction (Fig. 3).

Figure 3 shows two conceptual spaces, 'Private Design Workspace, PWD, and Shared one (Carrara and Fioravanti, 2002; Carrara et al., 2004).

Each actor, collaborating on the development of the design with a multiplicity of solutions bound to his specific skillset, operates in his own PWD using, as is normally the case, at least one software tool. For reasons of simplicity in this case, we assume that the architect and the engineer use a single software tool called, respectively, Toolarch and Toolceng. In general, application tools are heterogeneous.

In general, application tools have heterogenous data due to different internal data representations which rely to characteristies ones that we are the most important ones in the actor's professional domain. A file PTarch, produced by Toolarch using the Tarch representation can only be intepreted by other installations of the same software, so that it cannot be read by Toolceng, and the same considerations apply to the Design file Pceng produced by Toolceng.

Description of the MetaKaad prototype

To overcome this obstacle, we have defined for Toolarch and Toolceng its record structure in the XML format representation (DTDi with i {arch, ceng} in figure 3), and implemented two import and export drivers, which convert the data syntax, symmetrically between the two formats: on one hand Tarch and Tceng, on the other, DTDarch and DTDceng respectively.

A project consists of an XML file, available in the SDW. This is an instance of a project datum prototype, and 'virtually' [9] incorporates objects and characteristics produced by other actors. When an actor imports the project into his own PDW, the shared XML file is 'translated' by his Perspective. In other words, only those characteristics which are managed by him/her or related to his role in the design process, are activated. To these are added those which are specialist to his/her context, and which are neither useful to the other actors, nor shared at all with them. Once an actor has modified a project with his own tools, if s/he likes, a new version of the project (a new XML file) is exported to the SDW, and the other actors can import it. In figure 3, the label DTDs indicates the XML representation adopted for the Shared project data (Tab. 1). The shared XML DTDs scheme includes the DTDi representations (in figure 3 i {arch, ceng}) in the sense that an XML file which is valid for DTDi

[9] *Actually, objects, agents and characteristics remain in the IA's the various actors, only some of these are published in the SDW by means of links.*

is also valid for the shared DTDs scheme.

To complete the description of this architecture, the “Filter” and “Perspective” components, are described. Both these are defined specifically for each software tool in the MetaKaad environment. The Filter logically filters the information (in XML) of PXi Private Project of the i-th actor, and integrates it into the Shared Project, PXs, to generate a new version of the P’Xs project (export).

The Perspective interleaves the information coming from the Shared Project PXs, the specialist characteristics of the PDW of the actor-specific point of view (import).

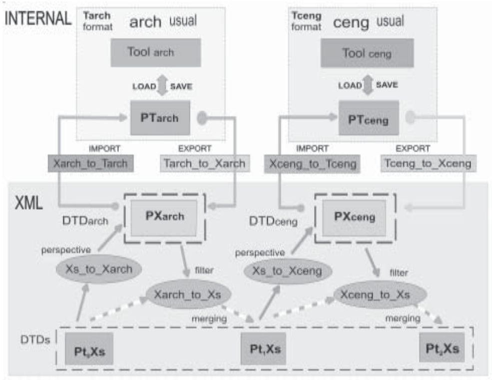

Figure 3. Interface between the ‘Private Design Workspace’ and the ‘Shared Design Workspace’ by means of XML.

```
<!ELEMENT PROJECT ( OBJECT )*>
<!ELEMENT OBJECT ( SLOT )+>
<!ELEMENT SLOT ( TYPES, ( SLOT+ | ( VALUE | LIST | CLIST
)+ ))>
<!ELEMENT TYPES ( TYPE+ )>
<!ELEMENT TYPE EMPTY>
<!ELEMENT VALUE ( #PCDATA )>
<!ELEMENT LIST ( VALUE | LIST | CLIST )+>
<!ELEMENT CLIST ( VALUE | LIST | CLIST )+>
<!ATTLIST PROJECT NAME CDATA #IMPLIED>
<!ATTLIST OBJECT NAME CDATA #REQUIRED>
<!ATTLIST SLOT NAME CDATA #REQUIRED>
<!ATTLIST TYPE APPL CDATA #REQUIRED
                CONT CDATA #REQUIRED>
```

Table 1. The DTDs syntax adopted for representing the Shared Project data in SDW.

As said, this generates a language whose strings are the projects being managed; each project has an identifier, ID, and is composed of a set of objects. Each object has an ID and a set of characteristics which define its place in the project. Each characteristic has, in turn, an ID and two distinct and separate parts:

• The first one indicates the set of software tools in the MetaKaad environment capable of managing it or interpreting its content. This part is composed of a series of Type tags, each of which contains the ID of the software tool it represents, and the attribute cont used to manage updates of the characteristic by the software tool itself;

• The second part defines the semantic content of the characteristic, which may be a simple value or be structured in a hierarchy of of subcharacteristics each of which has the same, or partually different, properties as its parent.

```
<!ELEMENT PROPERTIES ( PERSPECTIVES, ROLES, RULES,
COMMUNICATION)>
<!ELEMENT PERSPECTIVES ( PERSPECTIVE )*>
<!ELEMENT PERSPECTIVE EMPTY>
<!ELEMENT ROLES EMPTY>
<!ELEMENT RULES EMPTY>
<!ELEMENT COMMUNICATION EMPTY>
<!ATTLIST PERSPECTIVE NAME CDATA #REQUIRED
                CONT CDATA #REQUIRED>
```

Table 2. The DTDprop syntax in MetaKAAD environment.

The DTDs representation is accompanied by another, not indicated in figure 3 for reason of simplicity, to which is assigned the label DTDprop (Tab. 2) which defines the semantics of the data and the operations characterising the collaborative development of the project.

This grammar generates a language composed of strings of four sections each, as follows:

• ‘Perspectives’, is composed of as many Perspective tags are there are operational software tools in the MetaKaad environment; each fo these identifies a tool and contains a series of attributes (currently only the cont attribute) whose values are used for managing the data contained in the PXs project.

• ‘Roles’ defines the role concept for the various actors working on the project in the MetaKaad environment; depending on his role, each actor has a defined domain of authority in the development of the project and hence a set of relationships with the other actors with whom he is collaborating;

• 'Rules' defines the concept of process rules linking the various project activities run by the actors in the MetaKAAD environment;

• 'Communication' enables communications between the various actors.

Conclusions

The MetaKAAD prototype has been implemented, even if only partially, with a standalone application. The user is prompted for three pieces of information when he runs MetaKaad:

• the type of operation he wants to run;

• the type of software environment with which he wants to interact: that which another actor has used (import); or that which another actor will want to use (export);

• the name of the project.

Importation is used both for entering a new project and for updating a project already present in MetaKaad. It is curently possible to import only projects developed with Kaad; the AutoCad to MetaKaad import drivers have been only partly developed.

Exportation issues a copy of the project residing in MetaKaad to the user, and describes its characteristic data in the format suited to the selected environment.

From the above, it is clear that this mechanism will be seen to be a useful tool for interoperability of a group of projects and enabling diverse software applications, once registered with the MetaKaad environment, to interact straightforwardly. This will promote and enhance collaborative design between actors as they will each be able to work in their preferred and skilled way.

Acknowledgements

The research was partially funded by MIUR (Ministry of Education, the University and Research), Project of Research of National Interest 2002: "An integrated Product/Process model to support Collaborative Design in building".

References

Carrara, G, Fioravanti, A.: 2004, How to construct an audience in Collaborative Design - The Relationship among which Actors in the Design Process, in Rudiger, B, Tournay, B and Orbaek, H (eds), Architecture in the Network Society, 22th eCAADe Congress, 15 - 18 September 2004, pp. 426-434.

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 Pohl G. (ed.) Intelligent Software System for New Infostructure Focus Symposium, 30 July 2004, pp. 105-118. The Symposium was a field of 16th International Conference on: System Research, Informatics and Cybernetics, InterSymp – 2004, 29th July – 5th August.

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

Carrara, G. and Fioravanti, A.: 2001, A Theoretical model of shared distributed Knowledge bases for Collaborative Architectural Design, Strategic Knowledge and Concept Formation III Conference Proceedings, Sydney, 129-144.

Carrara, G, Fioravanti, A, Novembri, G.: 2001, Knowledge-based System to Support Architectural Design, in H. Penttilä (ed.), Architectural Information Management, Proceedings of eCAADe 01 Conference, 29-31 Aug. 2001, Helsinki, pp. 80-85.

Carrara, G., Confessore G., A. Fioravanti, G. Novembri.: 1995, Multimedia and Knowledge-based Computer-aided Architectural, in B. Colajanni and G. Pellitteri (eds), Multimedia and Architectural Disciplines, Proceedings of the 13th eCAADe Conference, Palermo, Italy, pp. 323-330.

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.

Borkowski, A., Branki, C., Palakz, W.: 2001, Towards collaborative creative design, Automation in Construction 10(5), pp. 607-616.

Kalay, Y.E.: 2001, Enhancing multi-disciplinary Automation in Construction collaboration through semantically rich representation, Porter, W.L. and Goldschmidt, G. (guest eds.), Special Issue: Design Representation, Automation in Construction, 10(6), pp. 741-755.

Cheng, N.-Y.: 2003, Approaches to design collaboration research, Automation in Construction, 12(6), pp. 715-724.

Garner, S. and Mann, P.: 2003, Interdisciplinarity: perceptions of the value of computer-supported collaborative work in design for the built environment, Automation in Construction, 12(5), pp. 495-500.

Hofstadter , D.R.: 1979, Gödel, Escher, Bach: un'Eterna Ghirlanda Brillante, Adelphi Ed., Milano, pp. 474-517. Italian edition of Gödel, Escher, Bach: an Eternal Golden Braid, Basic Book Inc.

Khemlani, L.: 2003 Should We BIM? Pushing the State of the Art in AEC, CADENCE, 18(6), pp. 6-12.

Kim, I., Liebich, T. and Maver, T.: 1997, Managing design data in an integrated CAAD environment: a product model approach, Automation in Construction, 7(1), pp. 35-53.

Laiserin, J.: 2000, Collaborative AEC Design, CADence, 15(11), pp. 10-17.

Laiserin, J.: 2001, AEC Collaboration, CADence, 16(6), pp. 10-17.

Pohl, J.G.: 2004, The evolution of Intelligent Computer Software and the Semantic Web, in Intelligent Software System for New Infostructure, Focus Symposium and InterSymp-2004 Conference Proceedings, Baden-Baden, pp. 11-34.

Pohl, K.J.: 2002, The Underlying Design Principles of the ICDM Development Toolkit, in Collaborative Decision-Support Systems, Focus Symposium and InterSymp-2002 Conference Proceedings, Baden-Baden, pp. 51-58.

Pohl, J. and Myers, L.: 1994, A Distributed Cooperative Model for Architectural Design, in G. Carrara & Y.E. Kalay, (eds.), Knowledge-Based Computer Aided Architectural Design, Elsevier Science Publishers, Amsterdam, pp. 205-242.

Rosenman, M A., and Gero, J S.: 1996, Modelling multiple views of design objects in a collaborative CAD environment, Gero, J.S., (guest ed.), Special Issue: Artificial Intelligence in Computer-Aided Design, Computer Aided Design, 28(3), pp. 193-205.

Tagliasco, V., Manzotti, R.: 2001, Coscienza e realtà: una teoria della coscienza per costruttori e studiosi di menti e cervelli, Il Mulino Ed.

Williams, A.: 2003, Principal Architects and Hands-on CAD, CADENCE, 18(8), pp. 8-14.

Woo, S., Lee, E. and Sasada T.: 2001, The multiuser workspace as the medium for communication in collaborative design, Automation in Construction, 10(3), pp. 303-308.