eCAADe18

State of the art versus state of practice
in computing for the design and
planning processes

PROMISE AND REALITY

Edited by
Dirk Donath
Bauhaus-Universität Weimar

A framework for an Architectural Collaborative Design

Gianfranco Carrara, Antonio Fioravanti, Gabriele Novembri

Keywords
Architectural Design Process, Collaborative Design, Knowledge Engineering, Dynamic Object Oriented Programming

Architectural design: present situation

Complexity of the building industry (and consequently of building design)

The building industry involves a larger number of disciplines, operators and professionals than other industrial processes. Its peculiarity is that the products (building objects) have a number of parts (building elements) that does not differ much from the number of classes into which building objects can be conceptually subdivided. Another important characteristic is that the building industry produces unique products (de Vries and van Zutphen, 1992).

This is not an isolated situation but indeed one that is spreading also in other industrial fields. For example, production niches have proved successful in the automotive and computer industries (Carrara, Fioravanti, & Novembri, 1989).

Building design is a complex multi-disciplinary process, which demands a high degree of co-ordination and co-operation among separate teams, each having its own specific knowledge and its own set of specific design tools.

Establishing an environment for design tool integration is a prerequisite for network-based distributed work.

It was attempted to solve the problem of efficient, user-friendly, and fast information exchange among operators by treating it simply as an exchange of data. But the failure of IGES, CGM, PHIGS confirms that data have different meanings and importance in different contexts. The STandard for Exchange of Product data, ISO 10303 Part 106 BCCM, relating to AEC field (Wix, 1997), seems to be too complex to be applied to professional studios.

Moreover its structure is too deep and the conceptual classifications based on it do not allow multi-inheritance (Ekholm, 1996).

From now on we shall adopt the BCCM semantic that defines the actor as "a functional participant in building construction"; and we shall define designer as "every member of the class formed by designers" (architects, engineers, town-planners, construction managers, etc.).

Our background

Our team has implemented a software prototype KAAD (Knowledge-based Assistant for Architectural Design) focused on the design process, seen from the point of view of an architect.

The development and implementation of data and objects, their hierarchical relationships, were considered a prerequisite for managing building parts and building elements as a designer normally does.

The aim of the software has been to provide architects with a knowledge of the relationships and links that always underlie every stage of architectural design.

The Building Object is conceived of in unitary terms as the fusion of the Spatial Domain and the Technological Domain, which are somewhat akin to the two sides of the same coin. The KB of Spatial Domain refers to the Building Units (Hospital, Outpatients Department, Diagnostic Department, etc.) and the Space Units (Ward, Bathroom, Operating Theatre, Corridor, etc.). The KB of Technological Domain refers to the Functional Elements (Vertical Partitions, Floor, Roof, etc.) and the Construction Elements (Windows, Brick wall, Doors, etc.).

All the objects that make up the KB are interrelated by constraints which are either hard (non modifiable and automatically activated) or projectual (modifiable and capable of activation by designer). The constraints implemented are: adjacency, communication, furniture, fire escape path, sanitary fixtures, estimate of quantities, etc.

The software does not verify projectual changes and choices applied to the project after the design activity has been carried out but in real time, during it. This is possible because the objects are dynamic, and because they almost always have default values. The possibility of continuously checking constraints and bills of quantities at any stage and state of the project makes effective mutual assistance between designer and software.

KAAD, unlike other software, does not favour or impose a top-down or bottom-up approach to the design activity. The architect can, as he/she likes, work in both directions, free to concentrate on the most important problem at that moment for the project, whether this is a functional one or one of technical-constructive detail (Carrara, Kalay, and Novembri, 1994).

The entire software programme and Knowledge Base have been implemented using dynamic objects.

This enables the actor to define new objects and new constraints, since it automatically re-generates the structure of KB by a re-parent of objects **on the fly** during the activity of design, without having to recompile the program.

With the help of some innovations KAAD has gradually evolved to KAAD phase II (Carrara, Confessore, Fioravanti, and Novembri, 1995).

Conceptual approach to ARchitectural COllaborative DEsign

Nowadays the activity of architectural design is increasingly marked by a growing number of actors called upon to co-operate in the realisation of building objects or complex infrastructures.

Every designer is often simultaneously engaged on different projects and part of different design teams. In the age of the Web, design teams in international companies can work 24 hours a day.

In their own studios the actors aim at realising an efficient organisation possessing the necessary flexibility to handle a large number of projects with relatively few alterations inside a well defined procedure.

While on one hand this process appears to be inevitable from certain points of view, since this type of approach now characterises design in many sectors, on the other hand there is a risk that the final results will reflect the typical faults of such an approach, and this will inevitably have repercussions on the construction phases and the life of the building.

In the light of the foregoing we plan to develop a support system for Architectural Collaborative Design (ArCoDe) able to handle the following problems, which are distinct but at the same time complementary:

- Multi-project support to assist the designer in his activity which today is simultaneously applied to various projects.
- Inter-operability support able to aggregate and communicate information, coordinate the contributions and goals of the different actors, and render the consequences of the adopted choices explicit to every actor
- Inter-application support that translates objects for the respective application-specific tool models, and maps objects for the respective model structures of a specific-domain (e.g. architecture, geotechnics, etc.).

Framework and devised tools

To realise a support system with all these characteristics it is necessary:

- to define KBs able to gather objects/concepts involved in the activity of design teams, the choices made by each actor, design constraints;
- to copy information contained in an actor's previous project KB to a new project KB;
- to allow every actor to perform some project choices and to check if these run into design constraints from general, specialistic, project KBs (prototype level);
- to check if project choices made by one actor verify project choices previously made by the same actor or by others (instance level);
- to improve communication among actors, and representation of KB's (data, procedures, choices).

This support tool we called Intelligent Assistant (IA) is composed by:

$$IA_{ip} = KAAD_i + KB_p + Perspective_{ip} + Filter_{ij} + Constraint_{ij}$$

Where i = i-team; j = j-team; p = project; $KAAD_i$ = specialist KAAD of i-team; KB_p = KB of specific project; $Filter_{ij}$ = filters between i-team and j-team; $Constraint_{ij}$ = constraints (or external $Goal_{ji}$) between i-team and j-team.

The objects/concepts and the performed choices contained in the KBs should also have an extremely sophisticated and self-explanatory description of the object able to describe to an actor of a different discipline-specific domain all the aspects inherent in its constitution, its use and the related consequences.

Conclusions - MetaKAAD

The consequences of such a model are:

- The actor can be either a designer or an IA;
- Redundancy, because data and knowledge reside in a Common KB, and in many Specialist KB's;
- Inconsistency, because only net-object local images are always consistent, the net-objects may be temporarily inconsistent;
- The system continuously checks consistency of KB's and DB's;
- Phases and states of the building object life cycle must be strictly defined;
- Net-encapsulation of object perspective characteristic.

Our aim in ArCoDe is to develop a software, which we have called MetaKAAD, able to detect conflicts on the basis of its knowledge (which may be increased by learning from example techniques, Simon, 1984). MetaKAAD operates between the intersection set and the union set of team $_i$ (= $actors_i$ + IA_i).

$$\bigcap_{1}^{n}(actor_i + IA_i) < MetaKAAD < \bigcup_{1}^{n}(actor_i + IA_i)$$

MetaKAAD could have this notation:

$$\bigcup_{1}^{n} w_i(\text{actor}_i + \text{IA}_i)$$

where w_i represents the weight it assigns to the team $_i$.

References

1 de Vries, M. and R. van Zutphen. **The development of an architects' oriented product model.** Automation in construction, **1**, 143-151, 1992

2 Carrara, G. A. Fioravanti, G. Novembri. **Towards a New Generation of Computer Assistant for Architectural Design: an existing scenario**. Proceedings of eCAADe Conference, 1989

3 Wix, J. (1997). **ISO 10303 Part 106, BCCM (Building Construction Core Model) /T200 draft**.

4 Ekholm A. **A Conceptual Framework for Classification of Construction Works**, ITcon **1**, 1996.

5 Carrara, G and Y.E. Kalay, (Eds.). **Knowledge-Based Computer-Aided Architectural Design.** Elsevier Science B.V., Amsterdam, NL. 1994

6 Carrara, G., Confessore G., A. Fioravanti, G. Novembri. **Multimedia and Knowledge-based Computer-aided Architectural Design**, Proceedings of the 13th eCAADe Conference, Palermo, Italy, 1995.

Gianfranco Carrara, Antonio Fioravanti, Gabriele Novembri
Dipartimento di Architettura e Urbanistica per l'Ingegneria
Università degli Studi di Roma "La Sapienza".
Via Eudossiana, 18 - 00184 Roma – Italy
fioravanti@uniroma1.it