

An Intelligent Assistant for architectural design studio

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

tel. +39-6-4458-5165; fax +39-6-4458-5186; e-mail carrara@uniroma1.it

Abstract

Statement of objects

Conscious design: explication of constraints and their consequences.

Collaborative design: interaction among KB's of operators involved in architectural design.

It seems by now fairly accepted by many researchers in the field of the Computer Aided Design that the way to realise support tools for the architectural design is by means of the realisation of Intelligent Assistants. This kind of computer program, based on the Knowledge Engineering and machine learning, finds his power and effectiveness by the Knowledge Base on which it is based.

Moreover, it appears evident that the modalities of dialogue among architects and operators in the field of building industry, are inadequate to support the exchange of information that the use of these tools requires.

In fact, many efforts at international level are in progress to define tools in order to make easier the multiple exchange of information in different fields of building design. Concerning this point, protocol and ontology of structured information interchanges constitute the first steps in this sense, e.g. those under standardisation by ISO (STEP), PDT models and Esprit project ToCEE.

To model these problems it has brought forth a new research field: the collaborative design one, an evolution of distributed work and concurrent design.

The CAAD Laboratory of Dipartimento di Architettura and Urbanistica per l'Ingegneria has carried out a software prototype, KAAD, based on Knowledge Engineering in the fields of hospital building and of building for aged people.

This software is composed by an Interface, a Knowledge Base, a Database and Constraints.

The Knowledge Base has been codified by using the formal structure of frames, and has been implemented by the Lisp language. All the elements of KB are objects

KAAD does not check constraints after that design activity has been carried out, but during the design activity, also if it has not been completed in each its parts.

This software is able to select and to analyse the instances created, and to understand the results of elaboration done, according to selected choices and activated constraints.

Our objects are to model the interactions among operators involved in building design, to develop a framework software, Meta-KAAD, able to gather heterogeneous data in existent KB's, manage specialised KB's and control conflicts, in order to decrease conflicts and costs from collaborative design viewpoint.

Keywords

Design Process, Collaborative Design, Knowledge Engineering, Dynamic Object Oriented Programming.



Present situation

Complexity of building industry (and building design consequently).

The building industry involves the highest number of disciplines, operators and professionals compared with

other industrial processes. His peculiarity is that his products (building objects) have the number of their parts (building elements) not much different from the number of the classes, in which building objects can be conceptually subdivided. Another important characteristic is that building industry produces unique products (de Vries & van Zutphen, 1992).

This situation is not alone but, au contraire, is spreading in the others industrial fields. In automotive industry, e.g. the production niches are successful. In the domain of computer firms we can see that high technological innovation produces continuous changes in products so that doesn't exist any more repetitive series (Carrara, Fioravanti, & Novembri, 1989).

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

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

The problem of an efficient exchange of data among operators and among software tools has not been resolved yet, and the fail of IGES, CGM, PHIGS, confirms this. The incoming STandard for Exchange of Product data, ISO 10303 Part 106 BCCM, relating to AEC field (Wix, 1997), seems to be too complex to put into practice for professional studies.

Moreover his structure is too deep, and conceptual classification inspires to it does not allow multi inheritance (Ekholm A., 1996).

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

The categories in building section appear to be ill-defined because every actor can be seen in turn as a client, a designer, a supplier (Fig. 1).

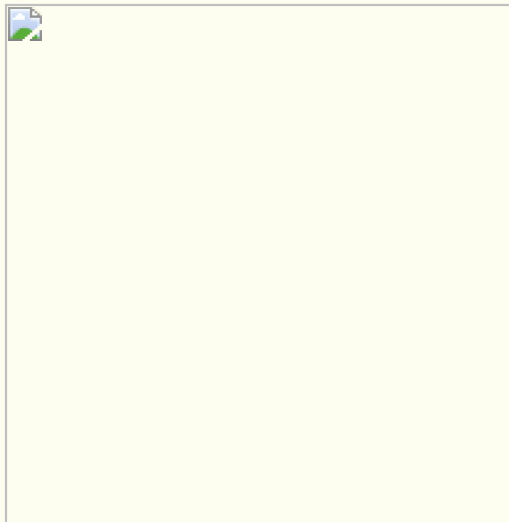


Fig. 1 - Exchange of roles in building section.

KAAD

Prof. Gianfranco Carrara is the head of CAAD Laboratory at the Dipartimento di Architettura e Urbanistica per l'Ingegneria. His team has carried out a software prototype KAAD (Knowledge-based Assistant for Architectural Design), focused on design process from a point of view of an architect.

The development and implementation of data and objects, their hierarchical relationships, were a need to manage building parts and building elements as designer usually does.

The aim of the software is to equip architects with knowledge of the relationships and links that always underlie every stage of architectural design.

The KB concerns hospital construction domain in particular an infectious diseases healthcare facility.

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

All the objects which form the KB are interlinked 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, estimate of quantities. The software does not take into account changes *after* the projectual activity has been carried out but *during* it: This is possible because the objects are dynamic, as we are explaining after, and because they have almost always default values. The possibility to continuously check constraints and bill of quantities at any state of the project realises an effective mutual assistance between designer and machine.

KAAD, unlike other software's, does not privilege, nor imposes a top-down or bottom-up approach to the design activity. The architect can, as he/she likes, works in both directions free to concentrate on the problem more important at that moment for the project, whether it is a functional one, or technical-constructive detail (Carrara, Kalay & Novembri, 1994).

Another main characteristic is the system of *views*. These are not mere changes of scale i.e. *dimensional rates of representations*, but also a change of meaning on the bases of the *context* conditions.

The context include information about the environment in which the designer acts: lifecycle phase; stage of design; scale of representation; symbolic views; state of the project and activated constraints (Fioravanti, Le Rose, Sgueglia della Marra, 1994).

For example when the system changes from a *wire frame view* to a *bubble diagram view* the system changes from a symbolic representation that points out the dimensions and dispositions of the walls to that of the hierarchical relations among Space Units.

We can define the Design Process as the Exchange of Information that interacts with the project (the Building Model) in a given Context.

$$DP = BM + EI (BM, C)$$

The system is composed by a Knowledge Base (goals and constraints and DB), a Geometric Modeller and a User Interface (Fig.2).

The KB is composed by: a Space Domain; Technological Domain; Shape Domain; Inference Engine. The Geometric Modeller performs solid modelling, its graphic primitives are in C++ language, is an hybrid edge type, and has the characteristic to have additive shapes (Kalay, 1989).

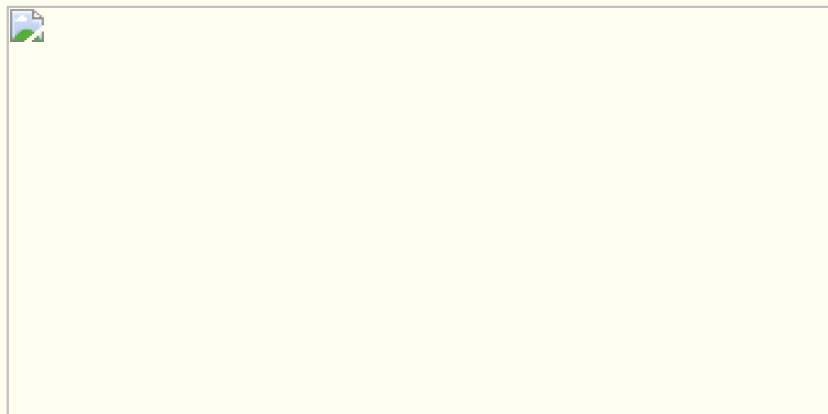


Figure 2. Structure of KAAD system.

The whole software programme and the Knowledge Base have been implemented by objects. The Knowledge has been codified using the formal structure of frames, and implemented by using the Lisp computer language.

We choose the Allegro CL/PC by Franz Inc. because it allows a Dynamic Object Oriented Design. That give the chance for the actor to define new objects and new constraints, since automatically re-generate the structure of KB by a re-parent of objects *on fly*, during the activity of design without recompiling the program.

As time passed KAAD has evolved to KAAD phase II by means of some innovations (Carrara, Confessore, Fioravanti, Novembri, 1995) (Fig. 3):

- Aggregate information, prototypes could be selected and aggregate, for needs of designer, by conceptual clustering techniques.

Multi-inheritance, prototypes could have two types of multi-inheritance: *virtual* when they do at level of prototype but not at instance level; *true* at level of instance and prototype;

Multimedia interface, prototypes could encapsulate video, pictures, e.g. a video of a patient room in a Nursing Space Unit prototype; a picture of several corridors in the Corridor Space Unit prototype;

Project KB, the project (i.e. an instance of KB) could have its project-specific KB besides general KB, e.g. new goals, new objects, new constraints

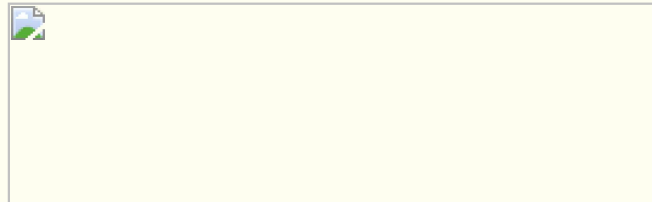


Fig. 3 - KAAD phase II.

- **Future developments**

The operating context

The activity of architectural design is today more and more marked by an increase of the number of the actors called to co-operate in the realisation of a building objects or complex infrastructures.

Such organisation often presupposes well-experienced actors of disciplines between them very different, every of which has necessarily often a partial and reduced point of view of the whole design process. The different actors increasingly often develop their activity to long distance, this is a virtual studio environment.

Every designer is often contemporary busy in different projects within different design teams and design teams in world companies can work 24 hours a day in Web age.

The actors aim at realising at their owns studios an efficient organisation provided by necessary flexibility for facing a great deal of projects with few alterations inside a well defined procedure.

If on one hand this process appears to be inevitable for some points of view, since this type of setting out characterises by now the design in a lot of sections, i.e. mechanical one; on the other it takes the risk that the final result puts in evidence the typical faults of this setting out, that will reflect inevitably in the phases of realisation and in the life of the building object.

In the light than exposed we consequentially think of realising a support system for design able to face two following problems distinguished but at the same time complementary:

- A multi-project support able to assist the designer in his activity that today is contemporarily developed on various projects.

- An inter-operability support able to aggregate and communicate information, coordinate the contributions and goals of the different actors, and make explicit to every actors the consequences of the adopted choices.

Before to proceed we try to better define some concepts involved in design environment

Concurrent design, when several actors act on the same object at the same time, their participation is strictly discipline specific. They transfer experience from a project to another, but not innovations. They are involved, in fixed categories, in many projects. They are "discipline specific minded", can be concurrent but competing also.

Cooperative design, when actors act not strictly at the same time and consequently not on the same objects (because these could be in the meantime changed). They transfer experience and innovation from a discipline to another within the same project. Sometime an actor supports somebody else and they are often jointly responsible. They are "project minded", may not be concurrent, but surely not competing.

Collaborative design, when actors act on the whole project at the same time, can be present many various disciplines (multidisciplinary teams). They transfer experience and innovations from a discipline to another and from a project to another. There is a responsibility widespread and loosened. They are concurrent and solid.

Multi-project support

We notices today a continuous increase of the specialisation of professional studios, engineering companies and professionals, that normally develop their activities within design teams for complex works.

If therefore it is difficult for an actor to hold in proper account within own activity, with one single project, the decisions performed by the other involved ones, this difficulty is enlarged by contemporaneous and different works.

The possibility to provide a designer with a tool able to individualise solutions already adopted in similar cases with the purpose to reduce time of carrying out his work, assumes a great importance. So the designer called to develop a specific job in a wider process, is provided with technical solutions already experimented and safe.

It is known that as in many professional studies and engineering company are predisposed for this goal, some repertoires of solutions considered efficacious and suitable for different contexts.

Very often yet these design solutions, extrapolated from the context in which they brought forth, they lose their efficacy and can become even dangerous when nothing is kept from the context in which they brought forth.

In this framework, a tool able to create these repertoires and at the same time able to store and to explain in immediate and intuitive way the reasons for which such solution has been taken, or define the similar situations that have been already studied, could represent a valid support for the designer for increasing the efficiency and the efficacy of his job.

Inter-operability support and inter-application

As the number of actors increase, it is needed always transmit in complete and exhaustive way a great amount of design information.

The diversity of the involved actors implies a difference of languages and points of view, different way to interpret same information that today is frequently reason of incoherence and losses of efficiency of the design process.

For this goal it is become necessary that design decisions assumed by every actor is transmitted in explicit, understandable, exhaustive, and consistent way to the other actors.

Achieving this objective is not yet easy inasmuch the mean by which this set of information is transmitted is very often represented by simple drawings. These succeed in transmitting basic information, but do seldom succeed in transmitting to the various operators what the motivations, the assumptions and the deep reasons of the choices that they have performed are.

Inter-application means that:

- application-specific model translates objects for respective application-specific tool;
- model mapping that translates objects for respective model structure of a specific-domain (e.g. architecture, geotechnics, etc.).

For the purpose to allow a true interoperability among actors involved in design activity, it would be desirable transmitted information were on one hand complete and exhaustive (self-explanatory), on the other were a "active tool".

In other words the information that one actor needs exchange and share, should be a KB able to be read from the point of view and from the most important goals of the others (objects with perspective characteristic). At the same time, interoperability support system should point out what choices conflict with constraints.

In this framework, that can be called web design or network-based design, the constraints are of two types: the first one already present in KAAD model, the second one imposed by other actors. Because a goal of an actor can be seen from another as a constraint.

The design, from this point of view, could be defined as a constraint import-export activity.

Methodological approach

To realise a support system of this type it is necessary:

- - To define a KB composed by: the objects/concepts involved in the activity of the design team, the choices taken place by every operators, the design constraints;
- To include information contained in the KB specific of one project already done by a single designer/actor;
- To allow every actor to perform some project and design choices and to check if these success in design constraints (general and specialistic);
- to check if project choices performed by an actor verify project choices previously performed by actor himself and by others;
- to improve communication among actors, and representation of KB's (data, procedures, choices).

This support tool, that we called Intelligent Assistant (IA) (Carrara, Novembri, 1995), can be seen like a KAAD evolution.

To put it into practice we have to introduce in the KB new objects/concepts types with these new characteristics:

- Net-objects
- -
 - Perspective, to change the aggregation properties of the same Building Elements by reason of the target actor, the state and phase of design process;
 -
 - Plug-in, the possibility to temporarily insert some application tool and some characteristics (slot) into prototypes from every actor;
 -
 - Filter, that translates the application specific tool data and the structures of the hierarchical models by switching on/off some characteristics;
- Net-constraint
- -

- Bi-lateral, e.g. the regularity and simplicity of frame designed by an engineer could not match with objectives of an architect.
-
- Multi-lateral, e.g. botany, ecology, energy saving, and bio-architecture (Fig. 5); insert a duct could do shear wall preferred to a column.
- Net-context
- -
 - The context depends not only from state and phase of the instances created by an actor, but of the whole (and distributed) project (according to definition of BNA the building process is divided into: feasibility - brief - structural design; preliminary design; final design; contractor and drawing; preparation; construction; usage).

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

The situation with several I.A. model, outlined in fig. 5, is:

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

Where i = i-team; k = k-team; p = project; $KAAD_i$ = specialistic KAAD of i-team; KB_p = KB of specific project; $Filter_{ik}$ = filter between i-team and k-team; $Constraint_{ik}$ = constraint (or external goals $_{ik}$) between i-team and k-team.

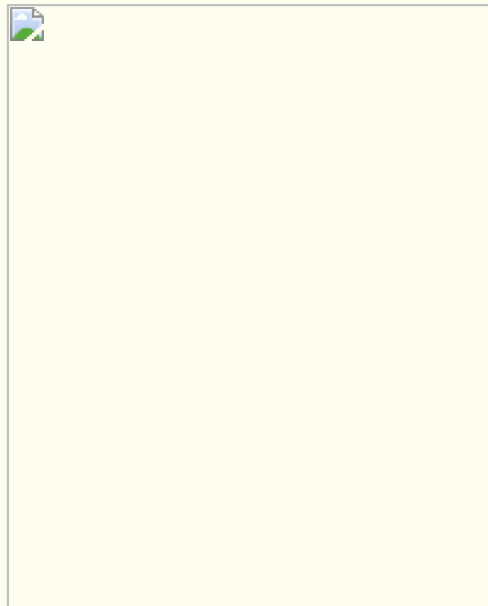


Fig. 4 - Actors and Intelligent Assistant.

Meta-KAAD

- The consequences of such a model are:
 -
 - The actor can be whether a designer, or an I. A.;
 -
 - Redundancy because data and knowledge reside in a Common KB, and in many Specialistic KB's;
 -

- Inconsistency because only net-object local images are always consistent, the net-objects are temporarily inconsistent;
-
- The system continuously checks consistency of KB's and DB's;
-
- Phases and states of building object lifecycle must be strictly defined;
-
- Net-encapsulation of object perspective characteristic.

The whole model of collaborative design framework, that we called Meta-KAAD, is outlined in fig. 5.



Fig. 5 - Collaborative design in a Meta-KAAD framework.

Differently from many researches that think should be one level of agents (Minsky, 1989; Petrovic, 1995) we think that should be a meta-agent or better, according to BCCM, a meta-actor who could manage the problem of multi-lateral network constraints.

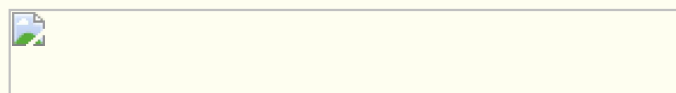
This is prefigured by Hofstadter (Hofstadter, D.R., 1988), when he spoke about the ant-eater and termites. Perhaps there is a hiatus between KAAD and Meta-KAAD; the problem is not the relationship among architects, quantity surveyors, contractors and manufacturers: this can be seen as an epiphenomenon at higher level.

In a conference about an oil-refinery project, the designer admitted that the design was very elementary because he have *only* had to mach few equipments for fractional distillation production...

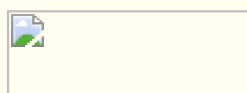
We have to study how juxtapose or move or rotate the actor + I.A. impulses and objectives like new mosaicists (movie 1).

[Movie 1](#). - Design activity is like to make a mosaic ... The whole is more than an addition of its elements.

The collaborative actor or Meta-KAAD should *only* minimise conflicts by his knowledge (or by learning from example) (Simon, 1984). Incidentally a simple way to discover emerging concepts for learning could be Meta-KAAD continually monitors the relevance of specific goals/constraints by means of statistical evaluation of instances created and of hits in net-related resources and objects/concepts. Meta-KAAD operates between the intersection set and union set of i-team's (actors_i + I.A._i).



Meta-KAAD could have this notation:



where w_i represent the weight it can assign to the i-team's.

- **Conclusions**

Prospect seems to be interesting and promising.

But it is already unresolved the problem about the global control of constraints and conflicts at Meta-KAAD level. Two are the possible scenarios:

on the former the system automatically optimises the weight (importance) of various I.A.'s, on the basis of considerate constraints and precedent instances that mostly resemble the context, the environment and the time of the project on which the teams are working. All the actors are at the same level. This way the collaborative design probably got a relative minimum of Conflict Function $f(c)$ (where c stands for constraints and context) (Fig. 6).

on the latter, like we stated for KAAD when an actor picked out a choice he did not mind of warnings that the system sent and of consequences in a short while, because he knew them will have been verified in a later stage, i.e., for Meta-KAAD, a meta-actor who selects I.A.'s and/or goals/constraints that have to be more taken into account. There is a meta-actor at a higher level. This way the collaborative design could reach an absolute minimum or a ill-defined relative minimum of Conflict Function $f(c)$ (Fig. 6).

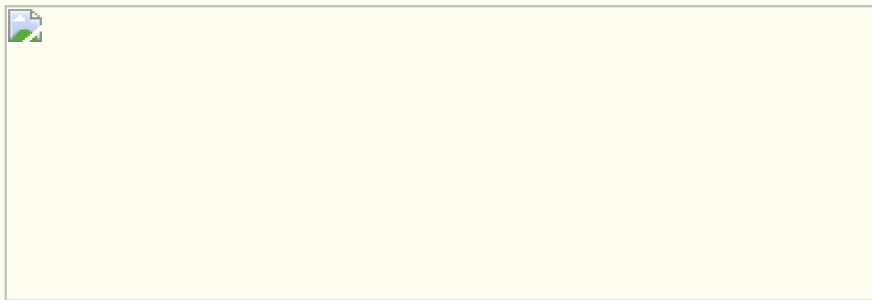


Fig. 6. The evolution of conflicts whit or without coordinator.

- **Acknowledgements**

The research has been partially financed by the Ministry of University, and of Technological and Science Research.

- **References**

Archea, J. (1987). "Puzzle-making: What Architects Do When No One is Looking", in Y.E. Kalay. (Ed.), *Computability of Design*, Wiley Interscience, New York.

Björk, B.C. (1989). "Basic structure of a proposed building model", *CAD*, **21** (2) March 1989: 71-78.

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

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

Carrara G., Y. E. Kalay, and G. Novembri. (1992). "Multimodal Representation of Design Knowledge", *Automation in Construction*, **2**, Elsevier.

Carrara, G, Y.E. Kalay and G. Novembri. (1994). "Knowledge-based Computational Support for Architectural Design". In *Knowledge-Based Computer-Aided Architectural Design*, Carrara, G and Y.E. Kalay". (Eds.), Elsevier Science B.V., Amsterdam, NL.

Carrara G, G. Novembri. (1995). "Computer Aided Intelligent Modelling in Architectural Design", *The future of CAAD*, Enschede, Elsevier,.

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

- Ekholm A. (1996). "A Conceptual Framework for Classification of Construction Works", *ITcon 1*.
- Fioravanti, A., L. Le Rose, C. Sgueglia della Marra. (1994). "KAAD: a didactical experience". *Proceedings of the 12th eCAADe Conference*, Glasgow, Scotland.
- Hofstadter, D.R. (1988), *Gödel, Escher, Bach: un'Eterna Ghirlanda Brillante*, Adelphi, Milan.
- Kalay, Y. (1989). "The Hybrid Edge: a Topological Data Structure for Vertically Integrated Geometric Modelling", *Computer Aided Design*, April
- Michalski, R.S., and Stepp, R.E. (1984). "Learning from Observation: Conceptual Clustering", in R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), *Machine Learning. An Artificial Intelligence Approach*. Springer-Verlag, New York.
- Minsky, M. (1989), *La società della mente*, Adelphi, Milan.
- Negroponete N. (1995). *Essere Digitali*, Sperlig & Kupfer, Milan, Italy.
- Petrovic, I. (1995). "A Framework for Cooperative Activities of Computer Design Agent". In L. Kalisperis and B. Kolarevic (Eds.), *Computing Design: Enabling, Capturing and Sharing Ideas. Proceedings of the ACADIA '95 Conference*, 19-22 October 1995 Seattle, WA, (171-186). Nittany Valley Offset, State College, PA.
- Schmitt, G. (1993). "Virtual Reality in Architecture", in N. Magnenat Thalmann, and D. Thalmann (Eds.), *Virtual Worlds and Multimedia*. John Wiley & Sons Ltd., England.
- Simon, H. A. (1969). *The science of artificial*. MIT press, Cambridge, MA.
- Simon, H.A. (1984). "Why should Machines Learn?", in R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), *Machine Learning. An Artificial Intelligence Approach*. Springer-Verlag, New York.
- Wix, J. (1997). *ISO 10303 Part 106, BCCM (Building Construction Core Model) /T200 draft*.

