

Needs requirements performances vs goals constraints values in collaborative architectural design

Gianfranco Carrara, Antonio Fioravanti

Dept. Architettura e Urbanistica per l'Ingegneria – University of Rome “La Sapienza” – Italy

gianfranco.carrara@uniroma1.it, antonio.fioravanti@uniroma1.it, www.dau.uniroma1.it

The following is a report on an on-going research study aimed at defining a support system for Collaborative Design in architecture. This paper concisely shows the latest results. In the present report the emphasis is laid on the relations among the *operators* and their different *goals*. We studied the preliminary phase of the architectural design, for which we defined a model and implemented a software system. The validity of the software system does not decrease when it is extended to the remaining design phases. The goal of the research is to improve collaboration among operators.

Collaborative Architectural Design, Knowledge Engineering, Design Goals, Constraints, Conflicts Reduction.

Antecedents

Collaborative Design (Woo, e al., 2001; Kvan, 2000; Gross et al., 1998; Jeng and Eastman, 1998) is a process characterized by a high degree of interdisciplinarity, by the delocalization of activities, by the segmentation of operations, by temporal appropriateness in the use of information, by the correct use of state of the art methods and technologies.

To manage these problems an alternative approach to the current market CAAD products is the one based on Knowledge Engineering proposed by many researchers (Carrara and Kalay, 1994; Kavakli, 2001). All these systems deal with a single model either of the design process or of the data, as in fact they possess either a unique Knowledge Base or an unique Data Base.

The aim of the research is to go beyond these limitations by having a network of several specialized KBs and developing innovative technologies (Carrara, and Fioravanti, 2001). Together these allow control over the whole architectural project on which a number of different networked operators are working simultaneously and remotely (specialists, designers, manufacturers, etc.).

Our Research Unit has been involved in Collaborative Design in the sense described above for a number of years (Carrara and al., 2001; Fioravanti, 2002). We have been dealing with these problems by means of Knowledge Engineering and relevant methodologies. The language used is LISP.

Needs Requirements Performance – Goals Constraints Values

In Design it is attempted to satisfy the Needs expressed by the client by means of an artifact (Simon, 1989). In order to become a design object *Needs* must be defined objectively by means of suitable specifications. The latter are defined as *Requirements*: the “objectifiable aspects of a need”. In fact any *need*, at least from the theoretical point of view, can be expressed by a set of *Requirements*. *Requirements* are characterized by performance specifications, test modes, range of acceptability, etc. The elements satisfying the *requirements* will provide the desired *Performance*.

These concepts, which are typical of design, have been formalized by means of objects (in the IT sense). Therefore the existing standards and the design objectives of an designer, in order to be represented in a Knowledge Base, need to be defined in one or more objects. These possess the characteristics of the aspects of interest.

In our Knowledge Bases *Needs* have been defined by means of prototype objects (including inferential engines) which we have called *Goals*; *Requirements* through sets of prototype objects; *Constraints*; *Performance* is measured through the *Values* taken on by the instance object.

Therefore the goal is a complex and structured set of *constraints* that applies to the whole Building (Building Object), i.e. when several designers and KBs are involved – such as costs -; while the constraint is a more limited problem inside one or two KBs – it usually determines the allowable values of a considered characteristic of a building component.

We define a *constraint* as “a functional of objective variables to be verified”.

Two different types of constraint are envisaged: ‘natural constraints’ or ‘design constraints’ (Carrara, Kalay, 1994). Natural constraints represent the limits of variation of characteristics deemed to be dictated by common sense.

The second constraint type is mostly represented by means of the inclusion in the prototypes of specific slots corresponding to the characteristics to which the *constraints* – CON - are to be applied. In the case shown in Figure 1 we give a simplified example of a goal representation and its dependencies. The number after the objects name defines it as an instance. The GOAL-034 is a set of CON-065 CON-055, CON-066. The CON-065 with the slot – ISA – (is a) defines its variable mother prototype, and with the slots MAX and MIN will allow the system to verify that the values taken on by a given characteristic are not greater or less than the given values. In turn the ISA-024 instance is defined by means of its values – Persp-014 - and – Persp-015- (Perspective). The To-Verify procedure will be used to check the constraint set. Note that in the instance – ATT-067 – (Attribute), the slot Measurement Unit - M.Unit - has no value, it remits to the instance – Persp-013 – (Perspective) that assigns a value in respect of the measure system of the country (for instance: meters or yards).

GOAL-034	ISA	Goal		Persp-013	Value	? ↓
	Description	description of the goal: Save Energy		To-Verify		if = ∀ Phase
	constraints	CON-065, CON-055, CON-066				& Country = USA
	To-Verify	Some procedures (CON-065, CON-055, CON-066)				Value= yd
						& Country = EU
						Value= m
CON-065	ISA	ISA-024, ISA-045		Persp-015	Value	? ↓
	Description	Decrease external surface or increase heat insulation		To-Verify		If = Preliminary Phase
	Attribute	ATT-067				& Actor = Architect
	MAX	Max Value				Value= H = 0.65
	MIN	Min Value				W/m ² °C
	To-Verify	Some procedures (...)		M.Unit		
ISA-024	ISA	CON		ATT-067	ISA	Attribute
	Description	Constraint			Description	modular height of the external wall
	Value	? ↓			Height	3.30
	To-Verify	? ↓			MAX	6.60
	Validity Value	Persp-014, Persp-015			MIN	2.40
					To-Verify	Control Procedure to calculate height between floors
Persp-014	Value	? ↓			Color	White
	To-Verify	If= Preliminary Phase & altitude > 1500 m or 1640.41 yd			Type	panel X
		Value= Surface/Volume < 4.5			M. Unit	Persp-013

Figure 1. Goal and Constraints: an example.

The characteristic values taken on by inserting object data represent the performances regarding the considered constraints. A design solution is such when the values of a characteristic are such as to satisfy constraints, and so the latter determine acceptable values for Goals. As has already been defined *Goals* have been calibrated to correspond to gradually more refined *Needs*.

To do this we introduced some innovations to the traditional ‘frame’: the changing of the object characteristic – the value - from a value to a pointer, which releases the frame structure from a preset depth of hierarchic levels to one of infinite levels; the changing of the objects characteristic - the slot – from a fixed datum to a situated one, which transforms the predefined structure and semantics of the object to variable ones.

Conclusion

The Goals and Constraints, through the described mechanism, allow the coherence of the design to be verified vis-à-vis the objective sets.

We think that this old-new object representation approach (LISP was born in 1959) and frames structure (McCarthy, 1960, Minsky, 1975, Winston, 1984), thanks to our innovations, can effectively manage the underlying explicit knowledge needed by the complex universe represented by architectural design.

The research in progress is revealing the potential of the approach adopted for the aspects linked to the preliminary phase of design through the first validations of the implementation of the software system described.

Acknowledgements

The research was funded by MIUR, Ministry of Education, the University, and the Research; National Research Project 2002: "An integrated Process/Product model for supporting Collaborative Building Design".

References

- Carrara, G. and Fioravanti, A.: 2002, 'Shared Space' and 'Private Space' Dialectics in Collaborative Design, *Collaborative Decision-Support Systems*, Focus Symposium and InterSymp-2002 Conference Proceedings, Carrara, G. and Fioravanti, A.: 2001, A Theoretical model of shared distributed Knowledge bases for Collaborative Architectural Design, SKCF '01 Conference Proceedings.
- Carrara G., Fioravanti A., Novembri G.: 2001. Knowledge-based System to Support Architectural Design. Intelligent objects, project net-constraints, collaborative work. *Architectural Information Management*.
- Carrara G. and Kalay, Y.E.: 1994, Past, present, future: process and Knowledge in Architectural Design, in G. Carrara and Y.E. Kalay (eds), *Knowledge-Based Computer-Aided Architectural Design*
- Fioravanti A: 2002, The relations among the actors and the architectural project in the design process: a rugby match, in Koszewski, K. and Wrona, S. (eds), *[Design-education]*,
- Gross, M.D., et al.: 1998, Collaboration and coordination in architectural design: approaches to computer mediated team work, *Automation in Construction*
- Jeng, Eastman, 1998, A database architecture for design collaboration, *Automation in Construction*
- Kavakli, 2001: modeling for detailed design process, from analysis to implementation, *Automation in Construction*
- Kvan T.: 2000, Collaborative design: what is it?, M, Special Issue eCAADe '97, *Automation in Construction*
- Simon, H.A.: 1996, *The Sciences of the Artificial*, MIT Press, Cambridge, MA, USA.
- Woo, S., Lee, E. and Sasada T.: 2001, The multi-user workspace as the medium for communication in collaborative design, *Automation in Construction*