Intelligent Structures for Collaborating with the Architect

COLAJANNI Benedetto, PELLITTERI Giuseppe, CONCIALDI Salvatore

Dipartimento di Progetto e Costruzione Edilizia – University of Palermo – Italy
http://dpce.ing.unipa.it

The number of different designers with different competencies collaborating in a building project is today conspicuous. An undesired consequence is the possible rise of conflicts between decisions taken independently by more than one specialist on the same building object. The early detection of such conflicts is then one of the most important features in collaborative design. Moreover, of great interest would be the possibility not only of automatic detection but also of solution proposal of at least the most manageable of those conflicts. In this perspective smart models of building components could be very useful. This is possible giving the building elements, represented as objects, the specific intelligence. A simple example of this possibility is given in this paper. In a precedent work we proposed a way of managing elementary spatial conflicts between building components tending to occupy the same spaces. The automatic detection derived from the previous declaration of two levels of constraints (soft constraint and hard constraints) in such way that a violation of them could be immediately signaled to the actor wanting to take the decision triggering the conflict. In this paper the topic is the consequences of the rise of a spatial conflict (occupation of the same space) between a column of a spatial frame of columns and beams, and another building object of any sort subject to a soft or hard constraint. The procedure identifies the minimum displacement of the two objects, propagates the column displacement to the other structural elements connected to it and checks the feasibility of the new configuration of the structural schema both with regard to the possible rise of new conflicts and with the compliance to previous structural criteria.

***Keywords**: Collaborative design, intelligent objects, conflict, constraints*

Introduction

Many architectural theorists (Zevi, 1948; Arnheim 1977) assume that the essence of the architecture is space, its configuration related to its use. Space is then the primary concern of Architects in designing buildings. Structure is secondary, unless it plays a role in the architectural image. The consequence can be a somewhat loose coordination between the architect and the structural engineer. Further, if structure has to remain unseen, the architect can design and draw a plan in which the structure (columns, shearing walls, etc.) is not marked. What is more likely if the collaboration between architect and structural engineer happens at distance, through electronic means of communication. And this is today becoming the prevailing professional paradigm. In this case collaboration is almost certainly asynchronous. The structural engineer has to cope with a fairly defined architectural plan to which he/she has to adapt the structure. What does not always happen easily since the structural engineer's view is different from the architect's one. Different is then the evaluation of the way in which a possible conflict between an architect's and a structural engineer's choice has to be resolved. Conflicts are resolved through negotiation. Negotiation means modification of one or both negotiating subjects decision in such a ways as to obtain consensus from both.

In a preceding paper (Colajanni, Concialdi and Pellitteri, 2000) we hypothesized that the architect's plan was not completely closed, blocked in all its part. Geometric constraints regarding position and dimensions of building objects generate the most relevant and frequent design conflicts. We assumed that this kind of constraint is not always absolute. Conversely there can be two types of constraints: soft constraints and hard constraints. Hard constraints are those the violation of which is strongly forbidden. Soft constraints are those that allow some variation of the restrained element both in position and in dimension. The violation of a constraint and, if soft, the domain of permitted variation must be encoded in such a way as to be immediately known to the subject, normally a specialist designer, that has generated the conflict with his/her decision. Early detection of possible conflicts is an important factor for a smooth flow of the design process. Explicit and timely declaration of constraints is one of the condition for this goal. In our paper we implemented a system of managing hard and soft geometrical constraints explicitly declared as such.

An efficient Design Assistant must not limit itself to aid in the conflict detection. It has also to aid in the search of solutions. This paper presents an attempt in that direction.

The Intelligent Structure

The situation taken into consideration is very frequent: an architectural plan in which the structure having no perceptual function in the designer intensions is not marked (see Figure 1).

Conversely hard and soft constraints (the last with their violation domain) are embedded in the CAD drawing. The structural engineer has to superimpose his frame of columns and beams respecting (see Figure 2), if possible the architects' constraints, or, if impossible, proposing the minimum of variations. An intelligent STRuctural

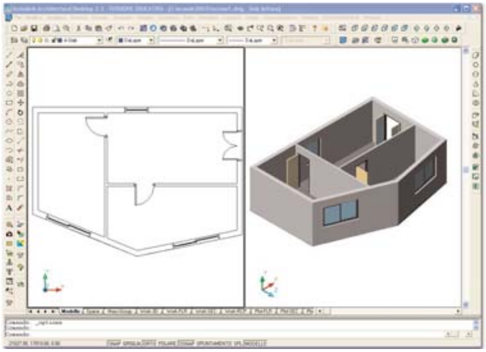

Figure 1. An architectural plan without structure previsions.

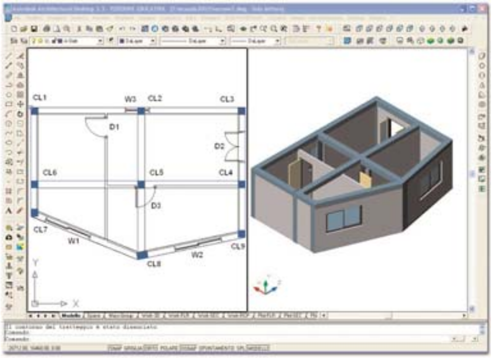

Figure 2. A structural plan superimposed to the architectural plan, an hard (CL2/W3) and a soft (CL5/D3) constraints are detected.

ASSistant (STRASS) can help him/her into exploring the state space of the possible solutions.

There are two ways of coping with the task of finding a structural frame consistent with the architectural plan. The first is tentative, empirical, locating the columns one by one trying to respect all the restraints and verifying “a posteriori” the possibility of the existence of a coherent solution. The second is to match the architectural plan with a complete proposal of the frame and operating the needed corrections in order to let the two fit.

This specific problem could be dealt with the classical tool of the problem solver (Lottaz, et. al., 2000) as it is possible to build up the representation of it in a continuous state space. Our pursued goal is not the solution of the problem in itself. It is the transformation of the structural subsystem into a set of intelligent objects (Anumba, et. al., 2002) with a smartness actually limited but able to grow. This will bring at the beginning a certain awkwardness, that, we hope, in the long run will disappear while the system is refining its capabilities. Committing the solution of the problem to intelligent objects of two categories column and beam implies a longer procedure for the specific problem but allows its inclusion in the general paradigm of the subsystem behaviour.

Another note. The proposed procedure comprises controls on the safety state of the beams the length of which are changed in consequence of the displacement of columns. An exact recomputation would entail in this moment a big complication. The actual state of the procedure is to be considered mainly as a test of the possible capabilities of entrusting to the structural assistant the exploration of the state space. At this moment a drastic simplification of the state space transforming it from continuous to discrete can be accepted. For each beam, besides the adopted section and reinforcement, a certain number of alternative instances, both stronger and weaker than the adopted one will be pre-computed. In step 4.2 if the test of safety on the beams is negative because of the change of length or charge, the most fit of the prepared alternative instances are adopted.

The core of the system is the smart columns. It is defined by means of its dimensions, position, beams concurrent on it. Its capabilities are:

- to be sensitive to the contact with constrained objects;
- to react to the perception of the constraints trying to transform the state of the whole frame from a conflicting to a non-conflicting one.

This second capability is the active one. In what follows we will explain the behaviour of columns and beams somewhat smart but not so much. In the sense that the process of finding a consistent solution runs in a sequential way with single steps each of which checks the existence of conflicts.

Before exposing the procedure some remarks are necessary. The problem we are dealing with can be represented in a state space. Its solution can be searched by means of general algorithms in the view of the min-conflict heuristics. Intelligence does not mean to posses all the capabilities enabling to deal with all the circumstances an intelligent object can meet. In this case it could mean to have the capability of calling to action most fit general algorithm. The sequential solution is only easier to implement. For this reason it has been chosen for the first attempt.

Let us examine the actions the column has to accomplish in order to exert its capability. If the column meets an hard constraint, it:

- 1.Individuates the position implying the minimum displacement consistent with the constraint.
- 2.Propagates the displacement to the columns connected to it by beams. Two cases are possible:
- 3.the displaced columns check automatically the consistency of their new positions to the

existing constraints:

- 3.1 no displaced column arises new conflicts; the procedure follows to step 5;
- 3.2 one or more columns arise new conflicts; the procedure re-starts from step 2;
- 4.no displaced column having arisen a new conflict; a control is made on the state of the beams the length of which is varied.
- 4.1 The control gives a positive result; the beams are still in safety conditions. The operation ends. Otherwise:
- 4.2 the control gives a negative result: at least one beam is no more in safety conditions. For each beam in that condition a test is made on the possibility of adopting one of the pre-computed alternative instances. If the test gives a negative result:
- 5.the column chooses the second possible position consistent with the hard constraint and restarts from step 2.

If the procedure does not converge STRASS declares its failure and calls for direct negotiation between the authors of the contrasting decisions.

In the case of a soft constraint the behavior of the system is almost the same. The only difference is a step preceding the procedure. The first action the smart column tries is controlling if the displacement the constrain allows is enough to leave the column in its place. If the displacement

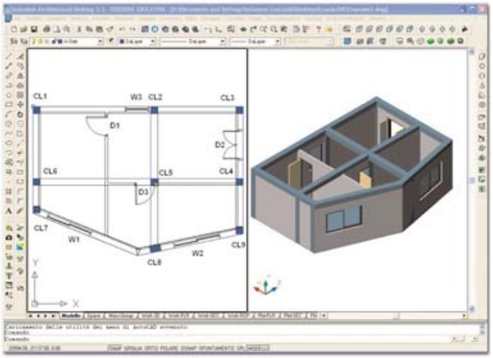

Figure 3. Hard constraints conflict resolved: C2 is displaced to the right border of W1.

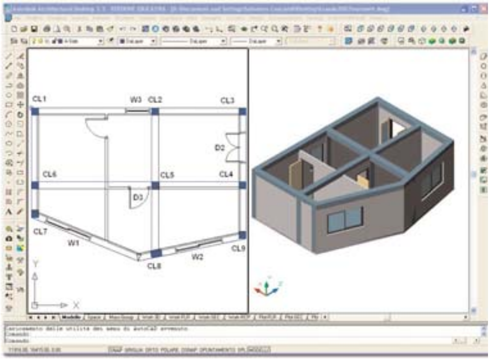

Figure 4. Soft constraints conflict resolved: D3 is displaced so as to live CL5 in place.

is not enough the sequence of actions starts from the position of minimum displacement of the column allowed by the new state of the constrained element.

Conclusions

Conflict resolution is an almost unavoidable problem that every design situation is called to face. The same can be said of the view of implementing the architectural elements as intelligent objects. Third, once again common is the problem of the adoption of a representation code allowing the transfer of data from one software to another. The basis for the solution of the last problem seems to be the IFC (Industry Foundation Class) standard. Some of the most important CAD already comprise a modulus allowing the exportation of a drawing from their format to the IFC one. This choice has consequences on the way of dealing with the second problem. Many attempts have been made of dealing with the problems of conflict solving starting from a specially designed representation code tending to the goal of easy communication among different specialists. Real Assistants to design problems, if they want to find a durable and consistent market must adopt a representation code universally accepted by the AEC community and this can be only IFC. It is important to remember that IFC is substantially an exchange format. It is not necessary that the software or the Assistants manage directly objects encoded in IFC format. It is enough that any software have an interface able to translate to and from other software data encoded in IFC code. Any attempt to find solutions of collaborative design problems by means of special representation codes is to be avoided if it is not compatible with IFC representation.

As it regards the first problem, indeed the idea of declaring constraints in an early state has advantages and drawbacks. The advantages are evident: knowing constraints allows the immediate detection of arising conflicts and this in turn allows negotiating and resolution. Just as evident are the fundamental drawbacks. Declaring a constraint means to declare that in case of conflict the integrity of constrained objects has to prevail on non-constrained ones conflicting with them. But it is difficult, and however tiring, to foresee all possible interferences between couples of objects. It can be usefully adopted in cases as the proposed one in which the nature of conflicts is clear an then constraints can be easily implemented.

References

Anumba, C.J.; Ugwu, O.O.; Newnham, L.; Thorpe, A.: 2002, Collaborative design of structures using intelligent agents, Automation in Construction, Vol: 11, Issue: 1, pp. 89-103.

Arnheim, R. (ed.): 1977, The Dynamics of Architectural Form: Based on the 1975 Mary Duke Biddle Lectures at the Cooper Union, University of California Press.

Colajanni, B.; Concialdi, S.; Pellitteri, G. : 1999, CoCoMa: a Collaborative Constraint Management System for the Collaborative Design, Architectural Computing from Turing to 2000 – proceedings of 17th ECAADE Conference, pp. 364-369

Colajanni, B.; Pellitteri, G.; Concialdi, S. : 2001, Un modello integrato per la gestione dei conflitti nella progettazione collaborative, L'Edilizia, 4, pp. 66-70.

Gross, M. D.; Yi-Luen Do, E.; McCall, R.J.; Citrin, W. V.; Hamill, P.; Warmark, A.; Kuczun, KS.: 1998, Collaboration and coordination in architectural design: approaches to computer mediated team work, Automation in Construction, Vol: 7, Issue: 6, pp. 465-473.

Kalay, Y. E.; Khemlani, L.; JinWon, C. : 1998, An integrated model to support distributed collaborative design of buildings, Automation in Construction, Vol: 7, Issue: 2-3, pp. 177-188.

Lottaz, C.; Smith, I.F.C.; Robert-Nicoud, Y.; Faltings, B.V.: 2000, Constraint-based support for negotiation in collaborative design, Artificial Intelligence in Engineering, Vol: 14, Issue: 3, pp. 261-280.

Zevi, B. (ed.): 1948, Saper vedere l'architettura, Einaudi, Torino.