Collaborative Design is a topic of particular current interest. Existing software allows a multiplicity of designers to work on the same project. What the software really allows is accessing to a part of the information of the project and changing it. Sometimes there is a hierarchical distribution of the power of change: some participants can be permitted to operate only on some limited layers of the object representation. In this case the changes they propose are to be accepted by a general manager of the design process. What is lacking in this kind of software is the explicit management on the reciprocal constraints posed by different participants. In this paper, an elementary Collaborative Design System is presented whose main concern is just the management of constraints. Each participant designs the part of the project of his/her concern instantiating objects comprised of geometric description, alphanumeric variables and constraints on both. Constraints can be of two types: absolute or defined by a range of allowed values of the constrained variable. A participant intervening later can accept the constraint, choosing a value in the permitted range, or decide to violate it. In this case the proposed violation is signalled to whom posed it.

**Keywords**: Collaborative design, design process, management system, participant designs, constraints violation.

Collaborative Design Systems of today

Collaborative Design is today one of the most frequented topics in the papers of the various CAD, CAAD, CAE and AI Conferences. This is of course because actually communicating through the Net is potentially the very paradigm of communication. Physical co-presence is no more the necessary situation to collaborate; it is quite a lucky exception. Moreover many are the approaches to collaborative design, but it seems possible to single out two main streams. The first is rather generic inasmuch as its task is not specific to architecture, also if it deals with architectural problems. Its task is to improve communication through exchange of information and possibility of all the participants to modify an unique object. The second tries to enter into the structure of the design process and from the understanding of it to derive the way the collaboration is to be organised. In any generic real design process there are phases of co-presence of the participants (synchronous phases) and phases in which they are distant in space and time (asynchronous phases). Hence the necessity has arisen to re-create the two corresponding types of collaboration: synchronous and asynchronous. Furthermore, another fundamental difference exists between collaborating actors: sharing the same
function (like two architects working together on the same design most often but not only synchronically) and collaboration between actors performing different functions (like the classic example of the architect and the civil engineer).

Yet another fundamental difference regards the representation method of the information on which agents collaborate: some ones argue for an unique model of the object to be designed then comprising all the information required by the various actors (Fruchter, 1966), some ones emphasise the differences between the views of actors performing different functions henceforth deriving the necessity of multiple modules each representing a view (Rosenmann & Gero, 1993 and 1997).

Eventually the possible cases can be schematised as in figure 1a.

Collaborative Design procedures

Any electronic multidisciplinary, collaborative design procedure should simulate or emulate the traditional pre-electronic (or, better, pre-Internet procedure).

Of course it is impossible to single out the multidisciplinary architectural design procedure, because, in fact, also if paradoxically, there are as many design procedures as the objects to design are. What is possible is to sketch a medium procedure, according to a probable logical hierarchy of disciplines, to be occasionally translated into a logical sequence of phases. Such a procedure is an abstraction but is a conceptual tool necessary to develop electronic ones.

A design procedure is naturally hierarchical but also iterative. Starting with an architectural model on which configuration of spaces and general perceptive aspects are delineated.

Proceeding with a structural proposal, adding structural elements such as columns, beams, shear walls, slabs, or interpreting and specifying with new functions objects already put in the model by the architect. The intervention of the mechanical and HVAC engineer follows, finding or introducing spaces for the services. Once the proposal of the architectural object is completed the project of the construction phase begin (see figure 1b).

It is not the case in this paper to examine the immense literature about the models and the programs about collaborative design. We will confine ourselves to the topic of conflicting constraints making a proposal of the way in which, according to us, it is possible to detect and resolve them.

Each intervention of a new participant to the design process generates feedback with the preceding ones because it can introduce, and often it does, conflicts that are to be detected and resolved.

The way in which this detection and resolution can be done depends, among other causes, on the way in which the model of the object to be designed is constructed.
Just two words about the choice between unique or multiple model. No doubt that the way in which different participants see the elements of a building is peculiar to each discipline. But what is important is the control of the ways in which an object characteristic that is not interesting for a discipline (and then could not be present in the relative model) affects another characteristic that reciprocally does not interest the first discipline and, in tour could not be present in its view. Generally it is thought that, in the case of a multiplicity of models what is common to all of them is the metric aspect of the object, and then any arising conflict can be reflected in it, but this not at all certain. For instance the density of a partition in concrete can be of interest both of the structural and the acoustical engineer but it is reflected neither in the metric model nor in the name of the material. It is rather difficult to define a priori the reciprocal dependencies. A possible representation could be the existence of an unique representation of each part of the object being designed containing all its characteristics, that could be questioned, imported and manipulated through disciplinary filters.

Conflicts detection

The question of detection of conflicts is the main concern of this paper. Our proposal is simple. Each participant that considers a decision pertaining to him as impossible to be given up declares it and highlights the element or even only a characteristic of it with a particular sign. In such a way a successive operator wanting to intervene on the marked element begins a negotiation with the participant that has posed the constraint. Of course in this choice, as in any other, there are advantages and disadvantages. The advantages are clear. The conflicts are not to be searched for through a checking of the consistence of the documents elaborated by the different participants. On the contrary they are immediately detected at the moment of their rise. Also the disadvantages are clear. The first of them is the growing heaviness of the representation. The second can be a tendency of each operator to declare as not to be given up the greatest possible deal of his/her decision.

A way to soften the impact of the imposition of constraints is to consider two types of them. One is binary. The constrain can be or integrally respected or integrally negated. For instance a window is in a certain position and of certain dimensions. Neither constraint can be modified. The window is or to remain unmodified or to be eliminates. Another type of constraint is less hard in the sense that is foresees a range of possible solution, the window can be displaced within a certain range of position an can be dimensioned within a certain range of dimensions. In this second case the proposition of the resolution can be directly found by the second operator.

A proposed procedure

We don’t want to propose a sequential procedure for the reasons already said. We suppose that in any case there will be a responsible of the project in the site of which at every moment of the design process, but the final one there will be a state of the model necessitating further elaboration. In each consolidated phase, components yet subject to constraints possibly interesting successive operators are highlighted in the graphical representation of the architectural object being designed. This is in order to point them up to whom in a following phase wants to intervene on them.

A generic dynamic phase of the design process is organised as follows:

- An operator that has not yet ended his work drives the model, possibly filtered by its view, on its site. He notices the possible existence of constraints on the components on which he wants to intervene. He verifies if the intervention he wants to do is consistent with the existing constraints. If it is, he makes the intervention adding parts or modifying them, sends the modified model back to the general project responsible informing him of the
intervention. The modified model substitutes the preceding one as representative of the state of the process.

- If the intervention is not consistent with the actual state of the model the inconsistency may regard one or both the types of constraints. If hard constraints have been violated the operator proposes all the same another solution representing it, practically another constraint, with the same code and informs both the general responsible and the participant that has posed the constraint. The negotiation starts.

- If the first constraint author accepts the modification he informs the general project responsible of his acceptance. The modified model becomes the starting model of the following phase. If the modification is not accepted the modifier is informed and the procedure starts again. Of course if the negotiation doesn’t give positive results the last decision pertains to the project responsible.

- If the constraint encountered is of the second type and the operator finds another solution within the range of allowed modifications, the constraint remains as it is possible that a successive operator has the necessity of adopting another solution always comprised in the range, but a further mark has to be put on it since the actual solution comes from a modification and a further change has to be negotiated with the operator that modified it.

This way of coping with constraints is apt for an asynchronous procedure, or, better for an asynchronous phase of a procedure. Asynchronous phases are unavoidable in procedures managing collaboration of different disciplines.

The implementation that we present is obviously only a primitive test of the operability of the hypothesis and does not pretend at all to be anything else than this. We think that this way of coping with constraints can be consistent with different programs managing collaborative design also depending on different approaches and is worth of further development. Once such a way of coping with conflicting constraints is accepted and given that the question regards only one phase of a far more general design process it is necessary to implement this chunk of procedure into a more complete Design Assistant. We intend to co-ordinate it with the Computer assistant devised by Carrara, Kalay and Novembri (Carrara G., Kalay Y. & G.Novembri, 1994).

The expounded procedure allows maintaining records of the project evolution. It is possible to maintain either only the accepted stages or those also documenting conflicts in order to record experiences about the way conflicts rise and are solved.

Another problem is the management of the intervention order by the different operators. It can be done through a priority order managed by the general
responsible and continually updated according the constraint rise.

At present, running under AutoCAD the programme comprises:

- Libraries of multi-layer walls, partitions windows, doors. Dimensions are parametric. Each type of component has attributes recording its main characteristics. The wall characteristics, for example, are: layers, composition, thickness, weight per square meter, conductance, acoustic isolation power.
- Procedures for adding new instances in the libraries.
- Procedures for instantiating components till the completion of a plan, representing the initial model of the building to be designed.
- Procedures for 3D representation to the model.
- Procedures of instantiating constraints of both types, hard, and soft.
- Chunks of programs for highlighting the constraints.
- Procedures moving or modifying the conflicting components thus modifying the model.
- Procedures for placing structural elements columns, shear walls, beams.
- Procedures for exchanging the model between the participants to the Collaborative Design.

**Conclusions**

We think that declaring the constraints instead of committing their detection to comparison of autonomously elaborated documents, or to the reactions of participants who perceive, examining those documents that their preceding design decisions have been violated, may shorten the time necessary
Figure 5 (right). Hard constraint (column C2). The proposed violation is signalled to whom posed it.

References


Benedetto Colajanni, Salvatore Concialdi and Giuseppe Pellitteri
Dipartimento di Progetto e Costruzione Edilizia
Università degli Studi di Palermo
Viale delle Scienze, 90128 Palermo, Italy
E-mail: bcolajan@unipa.it, ciesse@neomedia.it, pellitt@unipa.it