Individual and Team Contexts in a Design Process

Patrick Brézillon
LIP6, Case 169, University Paris 6
Patrick.Brezillon@lip6.fr

Abstract

The paper discusses the role of context in a collaborative design process, with a web-based environment for distributed architecture - engineering - construction teamwork. During this process, the decision is made collectively and progressively after a phase of negotiation among specialists from different domains. The goal of the project is to improve the teamwork performance by allowing participants (the specialists in different domains) to constitute a team and to collaborate in the project by interacting remotely in a virtual environment. The environment takes advantage of the current information technology to permit specialists to maintain their personal contexts and a shared team context, in which knowledge is acquired in concrete situations (i.e. knowledge is acquired in its context of use). The paper presents how context intervenes: (1) in the acquisition of the different types of knowledge as an intrinsic part of any negotiation, (2) how a decision making results of a progressive and negotiated building of a shared proceduralized context among the team participants, and (3) how participants in the teamwork can structure their own contextual knowledge (the individual contexts) and share, at least partially, their individual's tacit knowledge with other participants (in the team context). As a consequence, it emerges that design and learning are similar tasks, i.e. design is "learning by negotiating."

1. Introduction

Design is a complex activity because architects and engineers face ill-structured and incomplete problems. They constitute a joint team during all the lifecycle of a product, they process data, information, knowledge and the relationships among them in their own way, trying to stay compatible between them at the team level in order to reach the final joint decision. The design process is assimilated to a sequence of steps that go from abstract ideas to an embodied product. Such sequences include an uncertainty degree that decreases as the work progresses.

The two most common types of design in industry are adaptive design and variant design. These two types of design involve the use of known strategies or established design plans to reach new solutions. In such design types, decisional strategies for the problem decomposition and some classes of solution are already known. There is an initial perception of the product structure, and the disciplines, needed to solve the problems during the design, are identified. An example is the design of townhouses and apartment buildings. However, even in such situations, there is a specificity of each design that is expressed as constraints or contextual factors. This supposes to consider different types of knowledge in the design activity.

The design of an artifact (a product or a building) involves (1) the perception of its structure evolution during the design process and (2) the translation of each design state into the generation of new ideas. As captured by participants, the artifact attributes and their interrelations permit an organization of the design tasks based on the design state and the participants' role. This depends on the context in which the design evolves. Each person builds his own interpretation context of the artifact design by linking his specialized universe with the collective goal of his team. The artifact itself influences strongly the context in which participants identify and relate the tasks to be accomplished [8].

Hereafter, we discuss the role played by collaboration and knowledge in the development of computer systems aiming at supporting design teams during their interactions. Section 2 starts by reviewing some characteristics of knowledge related to the design practice. In section 3, the topics of collaboration and learning are explored and focused on the theme of computer-supported cooperative work. Section 4 presents briefly the application of the points discuss in previous sections (mainly the role of negotiation in design) in the web-based cooperative system – the SisPro project [3], [13].

2. Knowledge in context

Design is a knowledge-driven activity in which a description of the product corresponds to a set of requirements and constraints. Indeed, the design process involves a great amount of knowledge accumulated by engineers and technicians during their professional life. Knowledge sources have very different origins as formal education, individual experience and knowledge derived from similar cases. Thus, a cognitive structure for a design problem and its solutions is a knowledge-based learning process that must deal with very different types of knowledge as practical knowledge and theoretical assets.

2.1. Different types of knowledge

Knowledge is generally defined as the result of a process that has data as input. Data is a set of symbols mobilized by a person through sensors at a given moment when needed. An interpretation process then transforms data into information. The interpretation process relies heavily on the background knowledge of the person that interprets. Thus, information is data with meaning, something that is immediately usable and shareable by human beings on the basis of their knowledge [15]. However, an information presents an interest for a designer, if the designer is able to reason about and integrate this information into the body of knowledge in his background (i.e. establish relationships between the information and the knowledge already possessed).

Engineers and architects use basically two main types of knowledge when they design a new product: procedural and declarative knowledge. Procedural knowledge is generally expressed by procedures in organizational life, while declarative knowledge refers to more descriptive knowledge represented by equations and logical relations in books and manuals, or agents in new programming languages. This leads to a general discussion about the distinction between theory and practice, or “know how” versus “know that” [15]. Theory refers to domain models, causal explanations and rules that explain certain phenomenon, types of knowledge normally acquired by formal education. Practical knowledge refers to something that people acquire by doing efficiently a given task (human experts).

Organizational science considers explicit and tacit knowledge [14]. Explicit knowledge is easily shared whereas tacit knowledge is highly personal. Moreover, tacit knowledge can be divided into a part that can be made explicit and a part that cannot be made explicit, even if this later can be shared in a community of practice. This is the case for skills in craft jobs, and in teaching architecture and engineering where projects are assigned to the students.

2.2. Movements between tacit and explicit knowledge

Beyond discussions about the distinction between tacit and explicit knowledge, we think that any kind of knowledge can be made explicit or implicit depending on the circumstances, the persons and the society. In design, individual tacit knowledge must be shared and transformed (at least partially) in something understandable by other participants in the project.

Between *tacit* and *explicit knowledge*, Nonaka and Takeuchi [14] describes four types of exchange: socialization, externalization, combination and internalization, as represented on the Figure 1. Knowledge socialization refers to the creation of new tacit knowledge from shared tacit knowledge. Knowledge externalization refers to the conversion of tacit knowledge into explicit knowledge. Knowledge combination refers to the creation of new knowledge through the exchange and combination of explicit knowledge held by individuals in the organization. Knowledge internalization takes place when explicit knowledge becomes tacit, in a way similar to learning.

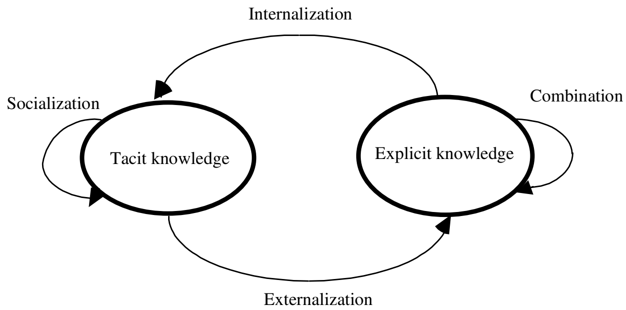

Figure 1. Movement between tacit and explicit knowledge

We think that the main contribution of Nonaka is beyond the distinction between tacit and explicit knowledge by the study of the exchanges between these two types of knowledge. The process of externalization is especially interesting as regards context use in design because it concerns the process of proceduralization that we introduced in [15]. This rises the question of why and when people decide to externalize. This question is particularly important in design because the development of an artifact supposes that know how has been previously captured. "Know that" may explain parts of the "know how" but is never sufficient to control complex process whence the idea of cooperative or interactive systems in which users work in interaction with a system. For example, mechanics laws can explain how and why bikes run but a bit of practice is necessary to ride a bike. This last example shows that the links between deep knowledge and practice are never necessary because many people ride a bike without any idea about mechanics.

If these movements between tacit and explicit knowledge are well known, Pomerol and Brézillon [15] give new insights on these differences by the introduction the explicit consideration of context.

2.3. Different types of context

From an engineering point of view, context is a collection of relevant conditions and surrounding influences that make a situation unique and comprehensible [2]. However, there are other points of views on context [4]. In the accomplishment of a task, a person identifies which knowledge is relevant to do his job based on his experience. Brézillon and Pomerol [5] call contextual knowledge these pieces of knowledge judged relevant and eventually mobilizable at a specific step of the decision making process (see Figure 2). A subset of the contextual knowledge at that step is invoked, structured and situated according to the focus corresponding to the step of the decision making process. This subset is called the proceduralized context.

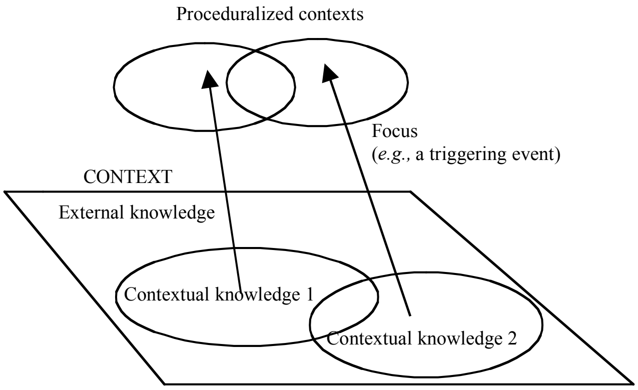

Figure 2. Contextual knowledge and proceduralized context

An important issue is the passage from contextual knowledge to the proceduralized context. This proceduralization depends on the focus on a task. Thus, this is task-oriented just as "know how", and it is often triggered by an event or primed by the recognition of a pattern. Another aspect of proceduralization is that the persons transform contextual knowledge into some functional knowledge or causal and consequential reasoning in order to anticipate the result of their own action. This proceduralization obeys to the necessity of having a consistent explicative framework to anticipate the results of a decision or an action. This consistency is obtained by reasoning about causes and consequences in a given situation. We can thus separate the reasoning between diagnosing the real context and anticipating the follow up. The second step needs a conscious reasoning about causes and consequences.

A second aspect of the proceduralization concerns a kind of instantiation. This means that the contextual knowledge or background context needs some further specifications to perfectly fit the task at hand. These precision and specification brought to the contextual knowledge are also a part of the proceduralization process.

In design, the reuse of knowledge abstracted from previous projects involves the building of an interpretation context of the former design situations in the light of the new situation. Thus, past results, or at least parts of them, could be ready for use or adapted for the current task. This point is considered in the section on negotiation in design (section 4.4).

2.4. Building of the proceduralized context

The proceduralized context represents some functional knowledge or causal and consequential reasoning. This description matches the situations experienced by engineers or architects when they have to look ahead to the consequence of their design decisions.

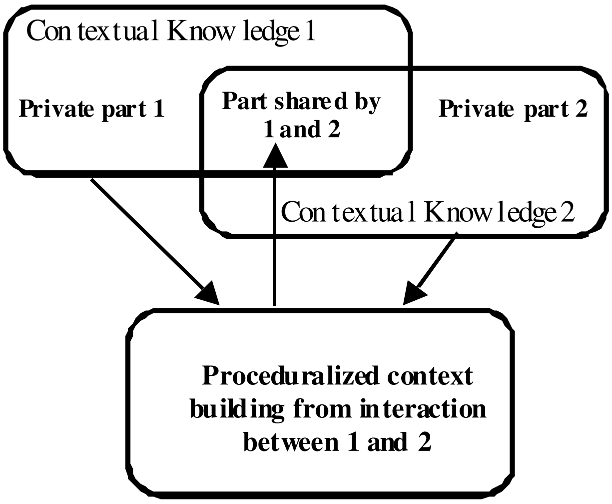

Figure 3. The process of building a shared conception of an artifact

The construction of the proceduralized context from contextual knowledge is often a process of communication in a community of practice, even if members of this community come from different domains. Figure 3 represents how the proceduralized context is built from contextual knowledge during the interaction between two persons. The interaction context contains pieces of the contextual knowledge that are introduced by a participant during the building of the proceduralized context in the focus of attention of all the participants. The pieces of knowledge are extracted from the body of the contextual knowledge of each person. They are jointly organized, structured and proceduralized by the two persons, and result in a shared chunk of knowledge. Generally, on the request of the other, a person may add some pieces of knowledge related to his previous utterance as a kind of explanation (similarly to the externalization process in [14]).

This progressive building of the proceduralized context is particularly important during negotiation. First, participants in the interaction accept any added elements. Second, the new pieces of contextual knowledge are linked, in a way accepted by all participants, to existing pieces of contextual knowledge in the interaction context. Third, the integration of the new pieces in the proceduralized context can be explained simultaneously in the team context and in each individual context of the specialists. Thus, the negotiation in design leads to a two-level agreement, in local contexts and in a global context.

If the final chunk of knowledge under construction is accepted by the participants, the knowledge pieces in the interaction context are integrated into a mutually acceptable knowledge structure—the proceduralized context, and then moved to their shared contextual knowledge when it gets off from the focus of the attention.

Thus, the proceduralized context contains all the pieces of knowledge that have been discussed, accepted and assembled by all the participants (at least with compatible assembling in individual contexts). This is a kind of learning by negotiating because each specialist integrates in his “know how” a chunk of knowledge containing arguments coming from other domains (see section 4.4 too). This new chunk of contextual knowledge (the proceduralized context at the previous step) may be recalled later as a whole, as any piece of contextual knowledge, to be integrated in a new proceduralized context. Thus, the more a person is experimented, the more the person possesses available structured knowledge. This explains why experts’ knowledge is more structured than novices’ knowledge.

This building of the proceduralized context is similar to some approaches in other domains. A key element here is the generation of explanations because argumentation corresponds to the construction of a shareable proceduralized context that supports the agreement of all participants in negotiation.

Clancey [9] discussed a view of the reasoning that involves forming a model of some system in the world in order to take action. The main argument of Clancey is that "the program's solution is not the name of a disease, but rather a causal argument having the structure of a proof, called the *situation-specific model*." A SSM is a model that is interpreted for the purpose of the tasks at hand, such as problem solving, diagnosis and repair.

Karsenty and Brézillon [10] show that if people must be able to explain their view for cooperating (and mainly negotiating), people also must cooperate for generating relevant explanations. Here the goal of the negotiation is not to have a unique and shared explanation, but to provide specialists with personalized versions of the explanation in their “languages,” versions that are compatible together.

Thus, the role of explanations in negotiation is:
- to do not intervene explicitly in the process but constrain it,
- to make explicit contextual knowledge, especially (a part of) the tacit knowledge of each specialist,
- to exhibit the structure of a proceduralized context in order to share it with other participants.

3. Collaboration in design

3.1. Learning and design

Learning and design are strongly related because the finding of a new solution involves generally the search and the acquisition of new knowledge, but mainly its integration in a body of existing and organized knowledge. Initial understanding and structuring of a design problem is obtained by (1) retrieving prior design cases and plans and (2) forming an abstract knowledge that can be further explored, in the spirit of case-based reasoning [11]. Thus, design and inductive learning are two similar processes that occur during interactions among agents. The areas of the communities of practice [7], [18] and in design science [8], [12] consider that both learning and design are social processes. This supposes that during such processes, there is an environment for negotiation and decision in which participants develop a shared understanding, knowing what is relevant to communicate and how to present information in a useful way.

3.2 A weak form of collaboration: Schemes and patterns

Beyond learning, it is interesting to make a parallel between design and cognitive ergonomics. The later domain uses the notion of scheme of action that we consider as a way to organize our knowledge in design. A scheme evolves according to different rules:
1. A scheme may be updated as a set of procedures that are relevant according to the singularities of the given situation.
2. A scheme may be enriched by addition of new strategies, i.e. by assimilation and by accommodation.
3. A scheme may be differentiated from an initial scheme to constitute a new one.
4. A scheme may be built as a combination of more elementary schemes.

Assimilation and accommodation are interesting in design. The assimilation process occurs for the use of a new tool from a scheme developed for a close (but different) use when the logic of functioning is different. Assimilation is our mind's way of saying, "This new information is already familiar to me." Some mistakes and some too fast generalizations may occur if one generalizes that the scheme is valid for several classes of situations. (For instance, a person may want to dry her cat in a microwave because it works with clothes.) The accommodation process modifies some parts of a known scheme to be tailored to a new tool. Different combinations are possible: combination of parts of different schemes, or combination of parts of a known scheme and new parts. The goal here is to reorganize the scheme to deal with a new class of situations. In such an approach, design starts from existing stuffs already accepted by everybody. Thus, negotiation is limited to secondary points in the design. We come back concretely on this point in section 4.4 of the Sispro project.

We observe also a similar position about design pattern for which the negotiation aims to learn experience as patterns. Patterns are an informal representation based on the features that are significant to the designer's everyday job. More specifically, patterns contain the result of years of experience, collaboration and refinement [16], not just abstract principles or strategies. A design pattern is a description of a problem and its solution in a given context. A pattern should document the problem, its solution and the consequences of using it. This permits to compare alternative solutions with full awareness of the consequences of each alternative. Design patterns capture the static and dynamic structures and collaborations of successful solutions to problems that arise when building applications in a particular domain. Design patterns are similar, at least in the spirit, with our notion of context when we consider a 3-uple (problem, context, solution) as represented by contextual graphs [6] to include in a uniform representation a procedure and all its variants called practices.

Thus, patterns help people to reuse successful practices (a successful solution is considered as a recurring solution to a standard problem, the meaning of pattern and of practice being different when contrasted with respect to a procedure): Patterns are a step toward handbooks for software engineers. People are concentrating on documenting the key patterns that successful developers use, but that relatively few developers thoroughly understand and consistently apply in their daily work. A pattern is a proven solution to a problem in a context. Here, context refers to a set of recurring situations in which patterns are applied. Alexander *et al.* [1] quoted that every pattern is formulated as a rule which establishes a relationship between a context, a system of forces which arises in that context, and a configuration that allows these forces to resolve themselves in that context. Thus, patterns are a way to contextualize knowledge and simplify collaboration in a static way.

3.3. Context-based interaction

A design team is a group of specialists in different domains with a common goal. Their main need is a functional support to improve their collaborative work. As any place to set up collective activities, the design arena is principally an environment for negotiation of constraints and decision making. The negotiation is hold in the team context, but arguments of the negotiation come from individual contexts but are converted to be understandable by all the specialists in the team context.

This is also a place where the identification of the tasks and the relationships between them allows to tackle the different domains and the role to be played by each participant. Thus, the routine issues addressed in supporting a design team are (1) how to manage and control the design information state, (2) how to share the correct information and (3) how to deliver design information in the correct time. This is a generic situation experimented by engineers, architects and technicians.

Social-cultural theory stresses that learning is promoted by social interactions among peers, and between peers and somebody more experienced. Learning takes place through verbal interaction and what is learned would be used when the learner tries to solve a similar problem independently. This view is grounded in Vygotsky's theories [17] that consider causal relationships between social interaction and the individual's cognitive development. Vygotsky suggests that by interacting (directly or indirectly) with more mature peers or experts, students can probably display abilities ahead of those displayed on their own. According to Vygotsky, the difference from knowledge derived from social interaction and knowledge achievable on individual's cognitive development defines the "zone of proximal development."

Thus, a collaboration process enables knowledge to be shared and allows the knowledge acquisition. Context is an important factor in the learning process because we must deal with the social and physical context in which learning takes place. This hidden factor must be revealed. As discussed above, context can be seen as a collection of relevant conditions and surrounding influences that make a situation unique and comprehensible. This is what is observed in design situations where collaboration is viewed as a process of building, negotiating and maintaining a shared conception of a problem under different aspects in which knowledge is learned in applied situations.

4. The SISPRO project

4.1. Introduction

The ideas presented in the previous sections are applied in the SisPro project [3], [13]. The SisPro project aims at the improvement of the design activity in the Architecture, Engineering and Construction (AEC) sector. It concerns a computer environment to aid a design team in a project and some experiments to evaluate the improvements achieved by the design team in real situations. The computer-environment part of the system aims at supporting collaboration activities and learning processes in a team. Experiments of the SisPro system were considered with professionals and with engineering students.

The SisPro project comprises a computer environment to support a team during the design of a project and during some experiments to evaluate the improvements achieved by teams in real design situations. It aims at supporting collaboration activities (including negotiation) and learning processes that are launched when working in a team. However, this concerns only a part of a whole "typical" workflow for building design. This part includes the main steps in the design of a building. SisPro's portfolio includes a computer environment that fulfills two main goals: (1) integrate relevant knowledge to design a building, and (2) develop teamwork skills among participants. The environment development follows some basic requirements: helping participants to structure contextual knowledge, permitting them to share their individual's tacit knowledge and proceeding knowledge acquisition through interaction.

4.2. The architecture of the SisPro system

The SisPro architecture (see Figure 4) takes advantage of client-server technology to allow virtual interaction during the project progression fulfilling the following demands:

- Allow participants to create, view or modify documents associated with a project,
- Allow participants to be aware of changes made in documents accordingly to their areas of interest, competence and role in the project,
- Support asynchronous argumentation and sharing of visual and textual information,
- Support synchronous discussion,
- Allow concurrent activities associated with a given task,
- Allow the identification, structuring and storage of contextual knowledge.

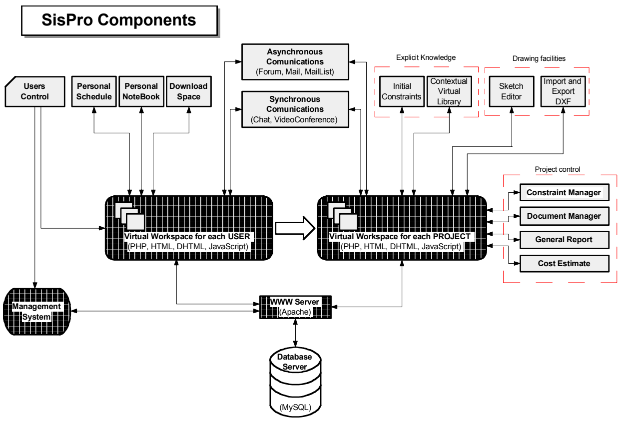

Figure 4. SisPro architecture [3]

The SisPro environment comprises a set of functions to enable synchronous and asynchronous communication and interaction. It includes two virtual workplaces. First, a virtual workspace is attached to each participant for defining their individual contexts. It permits the communication among all professionals that are on-line. Second, a virtual workspace for each project and common to all participants in the project defines the team context and allows controlled communication among a project team and provides a place for negotiation among participants.

4.3. Knowledge and context management in SisPro

Explicit knowledge, which is associated with building design and construction, is packed in context sensitive libraries. This permits participants to draw from the contextual knowledge the specific elements that will be used explicitly in the design of a building element. Such a library is the Architectural Contextual Library that was conceived by considering three main building blocks: architectural elements, design concepts and design solutions. The first building block is composed of the physical elements of a building, the second represents the organizing factors of the main elements with specific and typical situations and the third one presents some solutions that integrate the two previous ones.

Architectural elements represent the common knowledge on the functional elements of a building. This is the language that architects use for the specifications. This declarative language describes the general properties of architectural elements in a sharable way. These elements are represented in a common graphical language for spatial representation that is part of the architect repertoire. They are the invariant portions of a building (the contextual knowledge): every house has doors, windows, etc. Some architectural elements are grouped by function: structural elements (columns, beams, slabs), insulation elements (walls, roofs), horizontal and vertical circulation elements (stairways, ramps, corridors).

The second issue concerns the relationships between the architectural elements in applied situations. This concerns design concepts and embodies contextual knowledge. Contextual knowledge is related to constraints and requirements of each architectural element, and is associated with the spatial relations that must be established between them in order to have a functional integrity. Each architectural element can thus be studied in the different individual contexts of the specialists, and textual and visual media explain the interdependency between them. Such an explanation is built from the part of the contextual knowledge that is invoked and situated within the problem at hand, and the proceduralized context built at step of the decision process.

The third issue is a set of best practices that result from concluded projects. They reflect the articulation between the former two issues and are expressed in prototypical situations that are instantiated in assigned problems. As already stressed, the process of proceduralization requires reasoning about causes and consequences that are experimented in case-based reasoning. This module contains the design solutions and is composed of explanation texts, raster images and vectorial drawings.

When one compare Figures 3 and 4, one sees that the individual workspaces contain contextual knowledge of the participants (the individual contexts), and the project workspace is the place where is built collectively the proceduralized context corresponding to a step of the design process. The nature of the two types of context (the individual contexts and the team context) is different. Individuals contexts are essentially static and correspond to what participants bring in the design process. Conversely, the team context evolves dynamically along the design process and the progress of the negotiation among participants. At each step of the design, each participant agrees and the reasons, although rooted in different domains, are made compatible all together.

4.4. Negotiation in design

The building of a solution in design is similar to the building of a proceduralized context as discussed before. It is built progressively with the agreement of all the participants at each step of the building. A step of the solution building corresponds to the introduction of an element and the establishment of the links with the existing elements in the solution. At any time one can explain the rationale of the building. This point is particularly important when the participants in the negotiation come from different disciplines.

Negotiation intervenes in two situations. First, the solution is known and clearly identified. It generally results of the adaptation of an old solution. Adaptation may correspond to assimilation or accommodation (see section 3.2). Assimilation involves putting information into an existing scheme without changing the scheme. Accommodation is the process of changing our existing schemes in order to create new ones suitable to the new information or situation. In any case, the different specialists have already accepted the past solution. There is already a shared team context and the negotiation concerns minor changes for developing the new solution. Here, the negotiation concerns the adjustment of individual contexts of the specialist to make acceptable the solution by everybody.

Second, the solution is original and can not be drawn from a previous solution (e.g. use a new technology). Like for a proceduralized context, the new solution must be built one element after another one. In this situation, the team context is an arena where specialists must put items from their respective individual contexts into the team context in order to build the design solution as a proceduralized context. A new element is introduced by a participant and explained to the specialists in other domains. The main point here is to make compatible heterogeneous constraints in order to avoid conflicts. Here, the negotiation arises from the fact that, in parallel with the building of the solution in the team context, each specialist has a more or less clear model (in their domain) of what the new solution could be, and each "mental representation" (in their individual context) must be modified progressively according to the constraints of each specialist. In all the cases, this corresponds to a change in the knowledge of participants, either by introduction of external knowledge or by the rebuilding of past proceduralized contexts became contextual knowledge at that time.

In the spirit of the contextual-graph formalism [6], the first situation corresponds to an existing graph in which specialists plan to add a new practice (the design solution). Conversely, the second situation corresponds to the building of a new contextual graph, because there is no existing procedures, and the new solution will be assimilated to the procedure.

5. Conclusion

The introduction of web-based cooperation systems and tools represents a technology innovation in engineering education as well as professional practices. Companies are increasingly distributed worldwide, and project development is the result of a collaborative work between companies and their suppliers. This is reinforced by improved performance of Internet applications and decreasing costs in Internet connection.

Design and learning are social processes, in which social interactions that arise among participants promote learning. The design process encompasses a set of intellectual activities concerned with problem solving and negotiation processes among participants. Individual decisions are proposed, negotiated, validated and integrated with the collective constraints that dynamically change during the design progression. The design process is also recursive, especially in reason of the negotiation phase. There is a need in keeping traces of decisions in order to follow new routes or choose different conceptions. This is one of the features of SisPro, to permit the recording of the negotiation processes that take place and the assumptions considered in making a decision.

Seeking for information and learning is an intrinsic part of the design process, because new solutions often involve the search and acquisition of new knowledge pieces, or at least a new assembling of them (the proceduralized context). For example, a feature of the SisPro environment is a module called contextual libraries, in which contextual knowledge can be invoked and situated within the problem at hand.

The SisPro environment represents an effort in retrieving the old concept of "atelier", the traditional learning space in which people were bring together to engineering and architectural practice. It is a model to bring together the different disciplines that intervene in a building project taking advantage of information technology enabling people to interact and to collaborate in a common goal.

At a upper level, the duality between individual contexts, in which people develop their own experience as tacit knowledge, and the collective context, in which a global decision is negotiated in the respect of the individual decisions, can be pointed out in any domain now, not only in design. Making context explicit (separating, on the one hand, contextual knowledge and proceduralized context, and, in the other hand, individual and team contexts) is the way to help people working together.

Acknowledgment:
We thank the French Foreign Ministry that provides grants for the SART application.

6. References

[1] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., I., Fiskdahl-King, S. and Angel., S., “A Pattern Language”, Oxford University Press, New York, 1996.
[2] Anderson, J., “*Language, Memory and Thought*”, New Jersey, Erlbaum, 1995.
[3] Borges, M., Naveiro, R. and Souza Fillho, R., "SISPRO - A Computer Support System for conceptual design in architecture", *Proceedings of the 12th International Conference in Engineering Design*, Munich, Heurista, 1999.
[4] Brézillon, P., “Context in problem solving: A survey”, *The Knowledge Engineering Review*, 1999, 14(1), 1-34.
[5] Brézillon, P. and Pomerol, J.-Ch., "Contextual knowledge sharing and cooperation in intelligent assistant systems", *Le Travail Humain*, 1999, vol. 62 n. 3, , Paris, pp. 223-246.
[6] Brézillon, P., Pasquier, L. and Pomerol, J. Ch., “Reasoning with contextual graphs”. *European Journal of Operational Research*, 2002, 136(2): 290-298.
[7] Brown, J.S. and Duguid, P., "Organizational learning and communities of practice: towards a unified view of working, learning and organization", *Organization* Science, 1999, vol. 2 n. 1, pp. 40-57.
[8] Bucciarelli, L., "An ethnographic perspective on engineering design", *Design Studies*, 1993, vol. 9, n. 3, pp. 159-168.
[9] Clancey, W.J., "Model construction operators", *Artificial Intelligence*, 1992, vol. 53, pp. 1-115.
[10] Karsenty, L. and Brézillon, P., "Cooperative problem solving and explanation", *Expert Systems with Applications*, 1995, vol. 8, n. 4, pp. 445-462.
[11] Kolodner, J., "Towards an understanding of the role of experience in the evolution from novice to expert", *International Journal of Man-Machine Studies*, 1983, vol. 19, pp. 497-518.
[12] Naveiro, R. and Oliveira, V., “*O projeto de engenharia, arquitetura e desenho industrial”*, Juiz de Fora / MG, Editora da Universidade Federal de Juiz de Fora, 2001.
[13] Naveiro R.M., Brézillon P. & Souza F.R., “Contextual knowledge in design: the SisPro project”. Review Document Electronique, 2002 5(3-4): 115-134.
[14] Nonaka, I. and Takeuchi, H., “*The knowledge-creating company*”, New York, Oxford University Press, 1995.
[15] Pomerol, J.-Ch. and Brézillon, P., “About some relationships between knowledge and context”. Modeling and Using Context (CONTEXT-01). Lecture Notes in Computer Science, Springer Verlag, 2001, pp. 461-464. (Full paper at http://www-poleia.lip6.fr/~brezil/Pages2/Publications/CXT01/index.html)
[16] Schmidt, A., Takaluoma, A. and Mantyjarvi, B., “Context-aware telephony over wap”. Personal Technologies, 2000, 4(4): 225-229.
[17] Vigotsky, L., “*Mind in Society: The Development of Higher Psychological Processes”*. Cambridge, Harvard University Press, 1978.
[18] Wenger, E., “*Communities of practice*”, New York, Cambridge University Press, 1998.