

Explaining for Developing a Shared Context in Collaborative Design

Patrick J. Brézillon

Laboratoire d'Informatique de Paris 6, University Pierre et Marie Curie, France

Patrick.Brezillon@lip6.fr

Abstract

Collaborative design often meets together experts of different domains. Such groups need to share a context but often limited this one is often limited to few elements as common language, common tools and a same goal (the design process). Our goal is to developing a shared context to make compatible different experts' viewpoints. Making context explicit and shared supposes some explanatory capability. In this paper, we point out how context and explanation are intertwined, and the need of a uniform representation of elements of reasoning and of contexts for offering a larger spectrum of explanation in a collaborative design. .

Keywords: Explanation generation, shared context, contextual graphs, collaborative design.

1. Introduction

E-collaboration is generally defined in reference to ICT used by people in a common task [9, 10]. However, when speaking of e-collaboration, people seems to put more the emphasis on "e-" than on "collaboration", i.e. on the ICT dimension of the concept that on the human dimension. Along the human dimension, e-collaboration requires to revisit previous concepts of cooperation, conflict, negotiation, justification, explanation, etc. to account for the sharing of knowledge and information in the ICT dimension. We discuss in this paper of the explanation generation in this framework, thanks to making context explicit.

Any collaboration supposes that each participant understands how others make a decision and follows the progression of their reasoning to reach the decision. In a face-to-face collaboration, participants use a large part of contextual information to translate, interpret and understand others' utterances use contextual cues like mimics, voice modulation, movement of a hand, etc. In e-collaboration, it is necessary to find other ways in order to sent contextual information with the message. A way is to rely explanation generation on contextual cues [8]. Thus, explanation would play a role in e-collaboration more important than in face-to-face collaboration because there are types of explanation not possible in face-to-face collaboration. For example, some new concepts as the scrolling bar, the pop up menus and the information retrieval through names and

contents of files generate new ways to work. All groupware software do this.

Twenty years ago, Artificial Intelligence was considered as the science of explanation [11]. There are few concrete results to reuse now from that time.

A first reason concerns expert systems themselves and their past failures [7]. There was an exclusion of the human expert providing the knowledge for feeding the expert systems. The "interface" was the knowledge engineer asking the expert "If you face this problem, which solution do you retain?" The expert generally answered something like "Well, in the context A, I will use this solution," but the knowledge engineer retained only the pair {problem, solution} and forgot the initial triple {problem, context, solution} provided by the expert. The reason was to generalize in order to cover a large class of similar problems when the expert was giving a solution highly contextualized. Now we know that a system needs to acquire knowledge within its use context.

Another reason is that the user was excluded from the noble part of the problem solving because all the expert knowledge was supposed to be in the machine: the machine was considered as the oracle and the user as a novice [8]. Thus, explanations aimed to convince the user of the rationale used by the machine without respect to what the user knew or wanted to know. Now, we know that we need to develop a user-centered approach [3].

Capturing the knowledge from the expert, it was supposed to put all the needed knowledge in the machine, prior the use of the system. However, one knows that the exception is rather the norm in expert diagnosis. Thus, the system was able to solve 80% of the most common problems, on which users did not need explanations. Now, we know that systems must be able to acquire incrementally knowledge with its context of use.

Systems were unable to generate relevant explanations because they did not pay attention to what the user's question was really, in which context the question was asked. The request for an explanation was analyzed on the basis of the information available to the system.

Thus, the three key lessons learned are: (1) KM would stand for management of the knowledge in its context (context-based management of contextualized

knowledge in [5]); (2) any collaboration (including collaborative design) needs a user-centered approach (where the management of different contexts play a key role); and (3) an intelligent system must incrementally be able to acquire new knowledge and learn corresponding new practices. Surprisingly, we observe that the classical approach stays with stand-alone systems, and that collaborative design does not benefit of such lessons particularly helpful.

Focusing on explanation generation, it appears that a context-based formalism for representing knowledge and reasoning in collaboration design allows to introduce the end-users in the loop of the system development and to generate new types of explanations.

Now, new findings are available about context and allow a new insight on past problems abandoned previously by lack of a relevant solution at that time, like incremental knowledge acquisition, practice learning and explanation generation. Previously, they were considered as distinct problems. Now their integration in the task at hand of the user offers new options, especially for collaborative design.

Hereafter, the paper is organized in the following way. First, we present the background of our work by commenting briefly previous works on explanations in order to point out what is reusable, the relationships between explanation and context. Second, we discuss context management for explanation generation by providing a conceptual framework, our view of a shared context and a software called contextual graphs. In the third part, we come back on what is represented in a contextual graph, and propose different types of explanation and discuss potentialities in a collaborative-design process.

2. Background

2.1. Explanations in KBSs

The first research on explanations started with rule-based expert systems. Imitating a human reasoning, the presentation of the trace of the expert-system reasoning (i.e. the sequence of fired rules) was supposed to be an explanation of the way in which the expert system reaches a conclusion. Rapidly, it was clear that it was not possible to explain heuristics provided by human experts without additional knowledge. It was then proposed to introduce a domain model. It was the second generation of expert systems, called the knowledge-based systems. This approach reached also its limits because it was difficult to know in advance all the needed knowledge and also because it was not always possible to have models of the domain. However, the main weakness was the lack of consideration for the user and what the user wanted as explanation.

The user's role was limited to a role of a data gatherer for the system. A second observation was that the goal of explanations is not to make identical user's reasoning and the system reasoning, but only to make them compatible: the user must understand the system reasoning in terms of his own mental representation. For example, a driver and a garage mechanic can reason differently and reach the same diagnosis on the state of the car. The situation is quite similar in collaborative design where specialists of different domains and different geographical areas must interact in order to design a complex object. A third observation is that the relevance of explanation generation depends essentially on different contexts: the context use of the topic to explain [8, 1], individual contexts, the group context, etc. We discuss this point hereafter.

Even if expert systems of first generation are now abandoned, there are important results that we can yet reuse, such as the base for new explanations proposed in [13,14]. Thus, beyond the need to make context explicit, first in the reasoning to explain, and, second, in the explanation generation, the most challenging finding is that lines of reasoning and explanation are distinct and interact together, the line of explanation being able to modify the line of reasoning [1]. Thus the key problem for providing relevant explanations in a collaboration-design process is to find a uniform representation of elements of reasoning and of context.

2.2. Explanations and contexts

A frequent confusion between representation and modeling of knowledge and reasoning implies that explanations are provided in a given representation formalism, and the relevance depends on explanation expressiveness through this formalism. For example, ordinary linear differential equation formalism will never allow to expressing—and thus explaining—the self-oscillating behavior of a nonlinear system. Thus, the choice of a representation formalism is a key factor for generating relevant explanations for the actor and is of paramount importance in collaborative design with different experts and several tasks.

A second condition is to account for, make explicit, and model the context in which knowledge can be used and reasoning held.

3. Context and explanation generation

3.1. Conceptual framework

There is now a consensus around the following definition “context is what constrains reasoning without intervening in it explicitly” [6], which applies perfectly well in collaborative design (although with more complex constraints) where reasoning is developed collectively. From our previous works on context (e.g.

see [2]), several conclusions have been reached. First, a context is always relative to something that we call the (current) focus of attention of the actors. Second, with respect to this focus, context is composed of external knowledge and contextual knowledge. The former has nothing to see with the current focus (but could be mobilized later, once the focus moves), when the latter is more or less related to the focus (at least by some actors). Third, actors address the current focus by extracting a subset of contextual elements, assembling and structuring them all together in a proceduralized context, which is a kind of «chunk of contextual knowledge» (*a la* Schank [12]). Fourth, the focus evolving, the status of the knowledge (external, contextual, into the proceduralized context) evolves too. Thus, there is a dynamics of context that plays an important role in the quality of explanations.

3.2. Shared context

In a collaborative-design process, the shared context is the context of validity (or use) of the focus of the design. The shared context is built from contextual elements coming from the different experts' contexts. The shared-context building results from an incremental enrichment by contextual elements coming from the individual contexts of experts. Thus, a contextual element proposed by an expert will enter the shared context if it is accepted (validated) by other experts.

Each expert has a mental representation (i.e. his individual context) of the focus of the design and of its context of validity (the shared context). A contextual element provided by an expert must be integrated in other experts' mental representation, i.e. find a translation in their mental representations. This means that the collaborative-design process results by making the different views among experts compatible, and not necessarily identical because all mental representations are different. Explanations play a central role here for the integration of the contextual element in each mental representation. Because explanations are generated from existing contextual elements (the mental representation), the quality of the shared context (i.e. how context is made explicit) will, conversely, allow to improving explanation generation.

Explanation is a way to bridge each mental representation to the shared context. This supposes to be able to express the same thing by different ways because all the mental representations are different (e.g. for an expert in mechanics and an expert in electronic for the design of a car). Explanations must be specific for each expert, even if built from the same domain knowledge.

Each individual context contains an image of the proceduralized context that is built within the mental representation of the expert

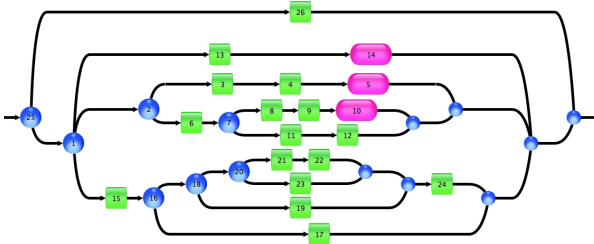
Finally a proceduralized context enters the shared context of the experts engaged in the collaborative design. Thus, the shared context will include the proceduralized context, its components (and their history) and the way in which these components have been put together, discussed, assembled, etc. However, each expert will finally retain in his individual context (his mental representation) an interpretation of the proceduralized context. All these interpretations, arising from the same proceduralized context that is built, will be compatible.

In collaborative design, explanation generation is a means to develop a shared context among the actors in order to have a better understanding of the others (and their own reasoning), to reduce needs for communication and to speed up interaction. The shared context results of a consensual view of all the experts because the shared context finds a specific translation in each mental representation.

As the context is associated with the knowledge, a context-based generation of explanations does not require an additional effort if such an explanatory knowledge is integrated in the knowledge representation at the time of the acquisition and representation of the reasoning (see [2] on this aspect). However, this supposes to have a context-based formalism allowing a uniform way to represent elements of reasoning and of contexts.

3.3. The formalism Contextual graphs (CxGs)

A contextual graph represents the different ways to solve a problem. It is a directed graph, acyclic with one input and one output [2]. Figure 1 gives an example of CxG. A path in a CxG corresponds to a specific way (i.e. a practice) for the problem solving represented by the CxG. It is composed of elements of reasoning and of contexts, the latter being instantiated on the path followed (i.e. the values of the contextual elements are required for selecting an element of reasoning among several ones). Elements in a CxG are actions (square boxes in Figure 1), activities (complex actions like sub-graphs), contextual elements (couples of circles in Figure 1) and parallel action groupings (a kind of complex contextual elements). A contextual element (CE) is a pair composed of a contextual node (e.g. large circles in Figure 1) and a recombination node (e.g. small circles in Figure 1). The number of paths between a contextual node (a large circle) and its recombination node (a small circle) correspond to the possible values (instantiations) taken by the contextual element. For example, in Figure 1, the CE "link target" can be instantiated by the value "HTML page", "PDF page" or "PPT page" (the three values are supposed to be exclusive).



- 25: Is the site already known?
 Yes 26: Look for new stuffs
 No 1: What is the link target?
 Html page 13: Open the target in a new window
 14: Activity-1
 PDF, DOC or PS page
 2: Is there a html version?
 Yes 3: Open target in new window
 4: Look for the keywords
 5: Activity-1
 No 6: Download the document
 7: Have I time now?
 Yes 8: Open the document
 9: Look for the keywords
 10: Activity-1
 No 11: Record document
 12: Close the window
 PPT page
 15: Open the target in a new window
 16: Duration of the download?
 Short 18: Is it for a course?
 Yes 20: Can page content be found?
 Yes 21: Copy the slide
 22: Paste it in a ppt doc
 No 23: Note idea for later
 No 19: Explore the presentation
 24: Go to the next slide
 Long 17: Close the window

Figure 1. Activity "Exploitation of an information" ([2])

Some mechanisms of aggregation and expansion provide different local views on a CxG at different levels of detail by aggregating a sub-graph in an item (a temporary activity) or expanding it. This representation is used for the recording of the practices developed by users, which thus are responsible for some paths in the contextual graph, or at least some parts of them.

4. Explanation in contextual graphs

4.1. What in a contextual graph?

A CxG represent a problem solving, a task to accomplish. The task or problem solving can be realized by different methods. For example, resolving an integral can be realized by different ways: integration by parts, change of variables, use of trigonometric variables, etc. The choice of one method over another one depends of contextual cues like the type of expression under the integral sign. This means that you learn general rules (a procedure), but the general rule must be contextualized in a practice to be applied effectively. CxGs addresses the representation

of practices with concrete elements corresponding to a situation in a given environment at a given time with particular actors, and, above all, used at least once by a user. The main interest here is that elements of a practice are acquired in their context of use (e.g. I use trigonometric variables because I think to be able to combine SIN and COS in a unique way).

Numbering of the elements in a CxG can be organized in different ways. In Figure 1, the elements are numbered in the order in which they have been introduced in the contextual graph. For example, the first CE has the number 25, when the following one has the number 1. This means that the first CE has been introduced in the contextual graph late during the development of the CxG. It is not important for following the development of a practice, but it is essential to know that the first element in the graph has been introduced after the following element if we have to explain a practice and contrast it over another one.

In the CxGs representation, practices are introduced incrementally. It is when the system, exploiting its knowledge in a CxG, fails to recognize a practice developed by a user that the system benefits of the situation (1) to acquire missing knowledge, and (2) learn a new practice. The introduction of a new practice appears as a specialization of an existing one. A specialization corresponds to a refinement of the context (and of the initial practice).

Regarding the conceptual framework initially described, a proceduralized context is an ordered sequence of instantiated contextual elements, but these contextual elements constitute a heterogeneous population. For example, the proceduralized context leading to the execution of the action 13 ("Open the link target in a new window") correspond to Value("Known site?") = "no" and Value("Link target?") = "HTML". Thus, if it possible to organize collection of contextual elements (e.g. domain, actor, environment, situation), it is not possible to structure, classify or categorize them. A proceduralized context is the unique way to organize contextual elements in a goal oriented way. This becomes an essential point for representing and integrating the elements coming from the different experts. Indeed, such contextual elements are considered actually as constraints, but these constraints are not represented explicitly in the collaborative-design process, and thus lost after for any explanation.

Another point is the importance to distinguish the CE from its instantiations. Two practices may differ by a CE or by the values taken by a given CE. This allows to pointing out that a situation that is normal in a context may become critical in another context. For example, Brézillon and Brézillon [4] explain that an important percentage of accidents on the road occur on a familiar way ("I cross every day this intersection since several year, and never I have had an accident". Yes, but this

time it was in a context never encountered before. The authors explain this by a “situation-dressing” approach that is complementary of the approach described here more oriented towards a description of the reasoning.

4.2. A large spectrum of explanations

In this context-based formalism of representation, we have established a typology of explanations, based on previous works and exploiting the capabilities of CxGs. By adding a new practice, several contextual information pieces are recorded automatically (date of creation, creator, the practice-parent) and others are provided by the user himself like a definition and comments on the item that is introduced. Such contextual information is exploited during the explanation generation. The richness of CxGs leads in expressiveness, first, of the knowledge and reasoning, and second of the explanations addressing different users’ requirements. The main categories of explanations developed in contextual graphs are:

- **Visual explanations.** They correspond to a graphical presentation of a set of complex information generally associated to the evolution of an item, e.g. the contextual graph itself, the decomposition of a given practice, the series of changes introduced by a given user, regularities in contextual graphs, etc.

- **Dynamic explanations.** They correspond to the progress of the problem solving during a simulation addressing questions as the “What if” question. With the mechanisms of aggregation and expansion, a user can ask an explanation in two different contexts and thus received two explanations with different presentations (e.g. an activity presented only by its name or from its content in terms of contextual subgraph). The dynamic nature of the explanation is also related to the fact that items are not introduced chronologically in a contextual graph. Finally, the proceduralized context along a practice is an ordered series of instantiated CEs, and changing the instantiation of one of them is changing of practice and thus changing of explanation.

- **User-based explanations.** The user being responsible of some practice changes in the CxG, the system uses this information to tailor its explanation by detailing parts unknown of the user and sum up parts developed by the user.

- **Micro- and macro-explanations.** Again, with the mechanisms of aggregation and expansion, it is possible to generate an explanation at different levels of detail. For complex items like an activity or a sub-graph, it is possible to provide on them a micro-explanation from an internal viewpoint on the basis of activity components. A macro-explanation from an external viewpoint is built with respect to the location of the activity in the contextual graph like any item.

- **Real-time explanations.** There are three types of such explanations. First, the explanation is asked during a problem solving when the system fails to match the user’s practice with its recorded practices. Then, the system needs to acquire incrementally new knowledge and learning the corresponding practice developed by the user (generally due to specific values of CEs not taken into account before). This is an explanation from the user to the machine. Second, the user wished to follow the reasoning of a colleague having solved the problem with a new practice (and then we go back to simulation). It is an explanation provided by the system to address an actor’s question. Three, the system tries to anticipate the user’s reasoning from its contextual graph and provides the user with suggestions and explanations when the user is operating. Such an explanation is co-built by the user and the system.

Moreover, these different types of explanation (and others that we are discovering progressively) can be combined in different ways like visual and dynamic explanations.

5. Conclusion

Relevant explanations are a crucial factor in any collaboration between human actors, especially when they interact by computer-mediated means like in e-collaboration. First, an e-collaboration loses some advantages of a face-to-face collaboration in which a number of contextual elements are exchanged between users in parallel with the direct communication. Second, a collaborative design can benefit of new ways to replace this “hidden exchanges” of contextual cues between actors by the use of the computer-means themselves. For example, it is possible to consider new types of explanation in a collaborative design.

A context-based generation of explanations is very promising for collaborative design because explanations use and help to maintain a shared context among actors. We show that making context explicit allows the generation of relevant explanations. Conversely, explanations are a way to make contextual knowledge explicit and points out the relationships between context and the design task at hand, and thus develop a real shared context.

A key factor for the success of relevant explanations is to use a context-based formalism that represent all the richness of the knowledge and reasoning that is in the focus. A good option is to consider context of use simultaneously with the knowledge. This supposes to have a context-based formalism like CxGs introduced in this paper. In such formalism, elements of reasoning and of contexts are represented in a uniform way. As a consequence, this allows developing new types of explanation like visual explanations, dynamic explanations, real-time explanations, etc. Indeed, we

have developed a new typology of explanations that include past works on explanations but goes largely beyond. Moreover, these different types of explanations are not independent and can be combined together to provide richer explanations.

However, this is only the first step. A promising path is to explore intelligent assistant systems. Now, there is a software piece called contextual graphs that is able to manage the phases of incremental acquisition and learning, and begins to provide some elementary explanations. A contextual graph represents the different methods that can be used for a given problem solving. Each path represents a practice effectively developed by a human actor (or a group of human actors). After a while, a CxG is a kind of corporate memory for this specific problem solving.

As a general learned lesson, expressiveness of the knowledge and reasoning models depends essentially on the representation formalism chosen for expressing such models. This appears a key element for collaborative design with multiple sources of knowledge and different lines of reasoning intertwined in a group work. This is a partial answer to our initial observation that collaborative design would be better understood if we consider jointly its two dimensions, the human dimension and the technology dimension. Then, explanation generation would be revised in order to develop “collective explanations” for all the (human) participants in the collaborative design, that is in each mental representation. Going one step further, it would be possible to compare with another view where ICT is controlled by an “intelligent agent” interaction in the collaborative design with human agents.

References

- [1] S., Abu-Hakima and P. Brézillon, « Knowledge acquisition and explanation for diagnosis in context », Research Report 94/11, LAFORIA, University Paris VI, Paris, France, 1994.
- [2] P. Brézillon, « Task-realization models in Contextual Graphs ». In *5th International and Interdisciplinary Conference on Modeling and Using Context*, Lectures Notes in Artificial Intelligence, Vol 3554, 2005, Springer-Verlag, pp. 55-68.
- [3] P. Brézillon, “Focusing on context in human-centered computing”, *IEEE Intelligent Systems*, 2003, 18(3): 62-66.
- [4] J. Brézillon and P. Brézillon, P. “Context modeling: Context as a dressing of a focus”. In: *Modeling and Using Context (CONTEXT-07)*, LNAI 4635, 2007, Springer Verlag, pp. 136-149.
- [5] P. Brézillon and J.-Ch. Pomerol, « Contextualized knowledge management ». *Proceedings of the 6th Workshop ISMICK-2008*, Rio de Janeiro, Brasil, 2008.
- [6] P. Brézillon and J.-Ch. Pomerol, « Contextual Knowledge sharing and cooperation in intelligent assistant systems », *Le Travail Humain*, 1999, 62 (3), PUF, Paris, pp. 223-246.
- [7] P. Brézillon and J.-Ch. Pomerol, “Lessons learned on successes and failures of KBSs”. Special Issue on Successes and Pitfalls of Knowledge-Based Systems in *Real-World Applications. Failures and Lessons Learned in Information Technology Management*, June, 1(2): 89-98, 1997.
- [8] L. Karsenty and P. Brézillon, « Cooperative problem solving and explanation », *International Journal of Expert Systems With Applications*, 1995, 8(4): 445-462.
- [9] N. Kock, R. Davison, R. Ocker and R. Wazlawick, « E-Collaboration: A look at past research and future challenges ». *Journal of Systems and Information Technology*, 2001, 5(1) : 1-9.
- [10] N. Kock, « What is e-collaboration? Editorial essay », *International Journal of e-collaboration*, 2005, 1(1): i-vii.
- [11] Y. Kodratoff, « Is artificial intelligence a subfield of computer science or is artificial intelligence the science of explanation? » In: Sigma Press, Bled. *Progress in Machine Learning*, 1987, pp. 91-106
- [12] R.C. Schank, « Dynamic memory, a theory of learning in computers and people », Cambridge University Press, 1982.
- [13] P. Spieker, « Natürlichsprachliche Erklärungen in technischen Expertensystemen ». Ph.D. Dissertation, University of Kaiserslautern, 1991.
- [14] W.R. Swartout and J.D. Moore, « Explanation in second generation expert systems ». In J. David, J. Krivine, and R. Simmons, editors, *Second Generation Expert Systems*, 1993, pp. 543 585, Berlin, Springer Verlag.