Context-based Assistant Systems: First Results

Les Systèmes d'Assistance basé sur le Contexte : Premiers résultats

Patrick Brézillon[1]

[1] Sorbonne University, Laboratoire d'Informatique de Paris 6 (LIP6), F-75005 Paris,
Patrick.Brezillon@gmail.com

ABSTRACT. The evolution of AI systems from expert systems, knowledge-based systems, joint cognitive systems, intelligent systems, intelligent assistant systems (IASs) and the coming generation of Context-based IASs (CIASs). CIASs require a deep focus on context and its relations with the users, the task at hand, the situation and the environment in which the task is accomplished by the user; the observation of users through their action and behaviors and not a profile library; a robust conceptual framework for modeling and managing context; and a computational tool for representing in a uniform way pieces of knowledge, of reasoning and of contexts.

RÉSUMÉ. L'évolution des systèmes d'Intelligence Artificielle depuis les systèmes experts, les systèmes à base de connaissances, les systèmes cognitifs associés, les systèmes intelligents, les systèmes d'assistance intelligent (SAI) et les SAI basé sur le contexte. Les SAIC requièrent une attention particulière sur le contexte et ses liens avec l'utilisateur, la tâche considérée, la situation et l'environnement dans lequel la tâche est accomplie par l'utilisateur ; l'observation de l'utilisateur à partir de ses actions et de ses comportements ; un cadre conceptuel robuste pour modéliser et manager le contexte ; et un logiciel qui permettent de représenter d'une manière uniforme les connaissances, raisonnements et contextes.

KEYWORDS. Context modelling, task realization, activity, contextual graphs (CxG) formalism, CxG-based simulation.

MOTS-CLÉS. Modélisation du contexte, réalisation d'une tâche, activité, formalisme des graphes contextuels, simulation basée sur les graphes contextuels.

1. Introduction

Artificial Intelligence (AI) was initially the field of formal research (General Problem Solver, etc.), but cognitive sciences invade AI, which, thus, concerns numerous disciplines where expertise (operational knowledge) is as important as technical knowledge, especially when there is no unique solution, rather one less bad than others. AI has been (and is always) a battlefield between "hard" science (i.e. theory-based science) and "soft" science (i.e. cognition-based science).

The expertise results of accumulation of problem solving where theoretical knowledge has been tailored to the situation at hand. Transformation of theoretical knowledge into operational knowledge is a process of contextualization during which actors adapt their knowledge to the task, the situation, and the local environment. Thus, operational knowledge is contextualized knowledge in actors' head. The expertise of such actors corresponds to find data in the world that can be interpreted in a relevant way (as information) to be associated with operational knowledge in actor's head.

However, organizations develop procedures from the accumulation of problem solving that are collections of secure action sequences to address a given focus in any case. For example, in our subway application, procedures are established by decontextualization of daily incident reports by operators in control rooms to cover a larger class of incidents. As a consequence, train drivers develop practices (i.e. their expertise) to adapt procedures to the situation at hand. The difference between procedure and practice is similar to the difference between theoretical knowledge and operational knowledge.

The need of exploiting actors' expertise for completing domain knowledge already in machines, is at the origin of the development of AI systems, starting with expert systems.

2. Lessons from the past

The separation of the representation of the declarative knowledge from its use is at the origin of first expert systems. The weakness is due to the fact that part of the knowledge comes often from technical domains but technical-knowledge organization and structure were not represented explicitly. Thus, operators introduced implicit knowledge to control the use of the knowledge (e.g. screening clauses to control the inference engine). The design of an expert system developed for *Electricité de France* was validated in a real extra-high voltage substation (400,000 volts) with success, but a second validation of the mockup in a different location was a failure: all the domain knowledge was the same, but contextual constraints (e.g. a narrow valley instead of a large place) lead operators to adapt the use of equipment to local conditions. A census in 1993 listed about 2500 expert systems, but the exact operational status is often not precisely stated, although more than 50% of the systems, which are installed in companies, were not used (Durkin, 1993).

The second important lesson is the need to acquire declarative knowledge as well as procedural knowledge. Procedural knowledge was often considered as a domain model in Knowledge Based Systems (KBSs). Knowledge was acquired in a monolithic manner beforehand its effective use. When a problem is detected with the knowledge in the system, the developer had to modify the knowledge base, re-load (re-compile or re-interpret) and then test the reasoning again. However, such problems were rather the norm than the exception because the domain model had to be applied in a variety of contexts that constrain in one way or another task realization. For example, the subways in Paris and Rio de Janeiro used the equipment of the same enterprise, but in different contexts. In Paris the subway is mainly underground and does not intersect with other means of communication while, in Rio, the subway is overground and some cows may cross the tracks. Moreover, there is air conditioning in the subway in Rio, not in Paris. Capturing and using knowledge with its context of use greatly simplifies knowledge acquisition because experts provide knowledge always in a specific context. The main lesson learned here is that task realization required to account for the role of the human actor (expert, user, etc.).

The abandon of Expert Systems or Knowledge-Based Systems was due to different causes. First, the user was considered as a novice in task realization, but systems was able to solve and explain 80% of problems, while users may solve unexpected situations adaptation or change of the known ways of solving the problem. Second, knowledge acquisition suffered a serious bias when the knowledge engineer questioned the expert that provided an answer as a triple (problem, context, solution) while the knowledge engineer retained only (problem, solution). It is like with opinion poll in which the question is written for a binary answer (yes, no) while the person surveyed would like to say “depends on…”. Three, the previous AI systems had never all the needed knowledge initially and the user could correct misunderstandings and problems. The role of the user was to be a data gatherer. The main point is this third cause is that AI systems and users jointly must intervene in the problem solving. Four, the introduction of an IA system in an organization often implies deep changes in the relationships of employees (the employee does not want to be spied on by his boss) and the sharing of power when different teams are concerned by the AI system development (the employee must deal with the computer engineer for correcting a bug in the system).

The lessons learned at this point are the need to have a user-centered approach and to deal explicitly with context. As a consequence, there is the emergent need of new methodologies for managing knowledge with its context. An approach (*Situated Cognition*, Suchman, 1987) insists on the importance to consider the environment in which we interact in problem solving, and the context at hand in organizational behaviors. Another approach (Joint Cognitive Systems, Woods, 1985) focuses on the complementarity of the competences and skills of the user and the system to exploit the asymmetry of their interaction during the problem solving. It implies that system and user are able to understand each other (more realistically, make compatible their interpretations) during a problem solving. This entails strong requirements about system design, such as the sharing of cognitive representations, an agreement about the contextual knowledge of the problem solving, the ability to follow each other reasoning, the capability of exchanging explanations, the capability to acquire incrementally new knowledge and learn new practices from the other, and a sharing of the interaction control.

AI systems rely on a logic of functioning often far from the logic of use. Complementarity of the user and the system implies that the system may contain either some knowledge (model or data) or functions (e.g. database exploitation) that really complements user's skills. This supposes that the system knows how to acquire knowledge incrementally with its context of use when needed. However, context must be made explicit. Conversely, humans usually function in an anticipative mode; operators explicitly check hypotheses to eliminate discovery or surprise. An anticipatory system would use knowledge (as a model itself and the relevant part of its environment) about future states to decide what action to take at the moment. Then, AI systems would become “intelligent system”, predict what will probably happen and adapt itself for the occurrence of a crucial or time-critical event. The anticipatory capability supposes that the system has a simulation component.

However, the line of reasoning of the user does not deal with the line of reasoning of the system like in early expert systems. Alignment of user’ reasoning and system reasoning is possible if the user and the system can explain their reasoning each other in the current context of the problem solving. Thus, explanations become an intrinsic part of the problem solving and an explanation may modify the line of the user’s reasoning as well as the line of the system reasoning. Context here is an extended version of the context in the previous approach because it also integrates live information on users, mainly on the basis of their actions on the system and on the real-world process to help the system to understand the user, not from a frame of reference but from how the user develop practices instead of procedures.

3. Towards context-based intelligent assistant systems

An intelligent system must be user-centered, that is, the system must apprehend the user “at hand” through his decisions, actions and behavior, not from an abstract user model or profile established in a library. Indeed, it seems better to consider an intelligent system as an intelligent assistant system. For example, a secretary solves tactical and operational problems to the boss and prepares elements for political and strategic problems. This view on the relationship between the user and the system refines the view in joint cognitive systems.

An intelligent system has to manage several interlocked tasks with the task at hand, such as acquisition, learning, and explanation. The multitask nature of intelligent assistant systems supposes to consider explicitly context in the representation of knowledge and reasoning. This is possible, thanks to context-based formalisms for modeling knowledge and reasoning. This new generation of AI systems develops contextualized reasoning like human develop practices. When the system fails, the user/expert can provide the needed knowledge and reasoning pieces in their contexts of use.

In our approach, context of the problem solving depends on the user’s focus. The focus divides context into external knowledge and contextual knowledge. The latter constitutes a kind of tank where contextual elements are related to the focus in a flat way, whereas the former has nothing to do with the focus at its current step. During the evolution of the focus, a subset of contextual elements is instantiated, organized, assembled and structured in a proceduralized context that is used in the problem solving at the current step of focus. The two key points here are (1) to consider context as composed of contextual elements, and (2) to distinguish contextual elements and their instantiations. In terms of decision-making, the selection of contextual elements is the qualitative part (or diagnosis part) of decision-making that analyzes the situation at hand and its context in order to extract the essential facts for the actions. Then, the instantiation of the selected contextual elements is the quantitative part (or action part) of decision-making.

We have developed a conceptual framework for context modeling and management that has been implemented in a piece of software called Contextual Graphs (CxG). CxG is a formalism of representation of the knowledge, reasoning and context in a uniform way. The CxG formalism is like logic or production rules, and a model developed in one formalism can be translated in another one. In the CxG formalism, a contextual graph is a model of all the known practices for solving a problem or making decision. As such, a contextual graph is a kind of experience base more on the way to solve a problem than on the problem itself (the goal of knowledge base). At the difference of previous knowledge bases built from procedures (or equivalent), a contextual graph concerns practices, i.e. the use of a procedure in different contexts. An intelligent assistant system can reason and accompany a user in the realization of a task on the basis of the experience bases.

A contextual graph is a directed acyclic graph with exactly one root and one goal. A contextual element is represented as a contextual node and a recombination node that gives a structure of series-parallel graphs to contextual graphs. The main lessons learned on the use of CxGs on several applications are (Brézillon, 2012):

- Contextual graphs are guidelines for activity management, i.e. task realization in contexts.
- An intelligent assistant system using an “experience base” will be able to explain, anticipate by simulation, and complete its knowledge on the task realization and alert about the variants abandoned along the practice.
- A system exploiting the CxG formalism will become increasingly “intelligent” because the CxG formalism allows an incremental knowledge acquisition and practice learning of new practices when the actor develops an original practice as a variant of an existing one.

An activity concerns the realization of a task, and thus relies on context. An activity, which exists on different branches of a contextual graph, is represented as a subgraph in the contextual graph. Another interesting possibility is to make compatible different viewpoints because a simple activity for one actor can result in a complex activity for another actor. In our application for the Parisian subway, the operator, which is responsible of a subway line in the control room, can ask to the driver of a train “Make empty your train of travelers at the next station”. The activity for the train driver is (1) stop at the next station, (2) make an announcement for requiring travelers to leave the train, (3) go and check that the train is effectively empty, (4) go back to the head of the train, (5) close and lock the doors of the train, and (6) leave the station to the garage.

Executive Structure of Independent Activities (ESIA) accomplish local goals. The order for executing independent activities does not matter (activities can be executed in parallel too), but both activities must be executed before to continue the development of the practice. In our application in medicine, histo-pathologists analyze images at microscope for identifying zones of interest from about 20 independent features. They give a personal weight to each feature (a parallel activity) by a system of vote (Brézillon et al., 2014). At the level of the contextual graph, an ESIA is a building block like an action or an activity.

The working context is the set of all the contextual elements in a contextual graph, their known values and their current instantiation. The working context defines the initial context of the task realization at the input of the contextual graph. After the crossing of the graph, the working context may be modified (generally by a change of the instantiation of a contextual element by an action during practice development). The working context plays a central role in the modeling of collaborative work, after an extension of the CxG formalism for managing iterative use of a contextual graph.

Collaborative work relies on a cyclic use of a contextual graph (that is directed) and the possibility to modify instantiations of the contextual elements in the working context. Modeling a collaborative work also supposes a simulation of the task realization. Indeed, this new function in the CxG formalism leads to a new way to model a task realization (by one or several users) by decomposing the task realization in elementary subtasks.

A CxG-based simulation (Brézillon, 2018) corresponds to the building of a sequence of turns with a mechanism for going from one turn to the next one. The turn mechanism is based on the notions of role, task, and the management of actor interaction with the task realization. The originality of a CxG-based simulator is to jointly develop and execute a practice.

The model of a group activity (including actor coordination) is a contextual meta-graph composed of a particular contextual element selecting the group member that will be the manager of the turn, and with as many branches in the meta-graph as members in the group. A member's activity is composed of a set of turns (i.e. several independent tasks) that are performed sequentially at different moments of a group-practice development. A turn is a local contribution of an actor to the group activity, and the group activity corresponds to a cyclic traversing of the contextual meta-graph.

4. Conclusion

The evolution of AI systems is linked to an increasing awareness of the role of context as soon as it is necessary to have a global view on the user, the task, the situation and the immediate environment, thanks to a modeling of knowledge, reasoning and context in a coherent conceptual framework. The main findings here is a description of a context as a set of contextual elements that can be instantiated in one way or another one. It is possible thanks to an operational definition of context and an implementation of the conceptual framework in a tool for analyzing knowledge and reasoning in context. It is the goal of the software "Contextual Graphs".

The exploitation of experience bases (i.e. contextual graphs) by AI systems is a new challenge for designing and developing context-based support systems able to tackle context in the same way as knowledge and reasoning. Moreover, the modeling of group activity by directed graph like CxG implies a cyclic use of the contextual graph, and this consequently implies a new approach of activity modeling in terms of elementary activities associated to the different group members. This new type of modeling also can be used for a unique user and thus represent successive trials of the user for problem solving.

Bibliography

[1] Bobrow, D.G., Brady, M.J. (Eds.): Artificial Intelligence 40 Years Later. Vol. 103, 370 pages (1998).

[2] Brézillon, P. (2018) CxG-based simulation of group activity. Review Modeling and using context, ISTE OpenScience, France (openscience.fr).

[3] Brézillon P., Attieh E. and Capron F. (2014) Modeling glocal search in a decision-making process. In : Gloria PhillipsWren, Sven Carlsson, Ana Respício, Patrick Brézillon (Eds.): DSS 2.0 – Supporting decision making process with new tehcnologies. Frontiers in Artificial Intelligence and Applications, 261, IOS Press, pp. 80-91. DOI : 10.3233/978161499399580

[4] Durkin, J. (1993). Expert Systems. Catalog of Applications. Intelligent Computer Systems Inc., PO Box 4117, Akron, Ohio 44321-117, USA.

[5] Suchman, L. (1987) Plans and situated actions. New York: Cambridge University Press.

[6] Woods, D.D. (1985) Cognitive technologies: The design of joint human-machine cognitive systems. AI Magazine, vol. 6, N° 4, pp. 86-92.