Investigating The Role Of Mutual Cognitive Environment For End-User Programming

Rémi Barraquand and Patrick Reignier

INRIA Grenoble
Rhônes-Alpes Research Center
655 Ave de l'Europe
38330 Montbonnot, France
{remi.barraquand,patrick.reignier}@inrialpes.fr

Abstract. In this paper we present a situated end user programming approach where user co-constructs, in an iterative process, a mutual cognitive environment with the system. We argue that co-construction of a mutual cognitive environment, between both the human and the system, is a key toward social human-computer interaction. Preliminary results are illustrated with a step by step case study: a user teaches the system new perceptual and abstract concepts using hand gesture and an interactive learning table.

Key words: End-User Programming, Situation Models, Cognitive Environment

1 Introduction

Information and communication services pervade nearly all aspect of our daily life, however they have no ability to perceive and understand the social situation or affective reactions of human. Furthermore, they have no sense of social roles played by interacting humans: systems are unaware of human goals and intentions.

Such abilities are beyond systems constructed with current technologies for two main reasons. First, systems lack an ability to acquire the skills for polite interaction using the implicit feedback provided by humans during interaction. Second, systems even lack the most rudimentary understanding of the social order that structures normal human interaction.

Common sense is the collection of shared concepts and ideas that are accepted as correct by a community of people. Social common sense refers to the shared rules for polite, social interaction that implicitly rule behavior within a social group [1]. To a large extent, such common sense is developed using implicit feedback during interaction between individuals. Reeves and Nass demonstrate convincingly in The Media Equation [10] that interactions with computers are identical to real social relationships and they argue that a social interface may be the truly universal interface. Thus our goal in this research is to develop methods to endow an artificial agent with the ability to acquire social common sense using day to day interaction with people. Our approach focuses on a key aspect of social common sense: the ability to build an appropriate mental model of social situations. We believe that such methods can provide a foundation for socially polite man-machine interaction, and ultimately for other forms of cognitive abilities.

Common knowledge is a phenomenon which underwrites much of social life. In order to communicate or otherwise to collaborate, humans make use of socially shared conventions, common understanding or background knowledge. While theory of mutual knowledge has the characteristic to produce a regression at infinity, Sperber and Wilson [12] developed a weaker but empirically more adequate concept, the mutual manifestness. In this paper we investigate the role that a mutual cognitive environment plays in collaborative social interaction and in the construction of situation models. We believe that building correct social situation models is a key toward designing context aware systems.

2 Related Work

Designing a context aware application is a complex task. There is no widely accepted definition of what is a context. For Dey et al. [4], context is “any information that characterizes a situation related to the interaction between humans, applications and the surrounding environment.” Dourish [6], on a more conceptual approach, proposes two categories for context models : representational and interactional. In a representational approach, context and activity are separable. Context is information about the environment where the activity takes place. The main question is how context could be encoded and represented. This is a more “programmer” point of view. In the interactional approach, context is a relational property between objects or activities. It is a more user centered approach. Dourish opposes those two points of view. We believe that those two approaches can be complementary. A software programmer, can use a representational context model to specify a first “general” contextual behavior (not adapted for everyone). The end user then adapt this general contextual model to its own needs. This adaptation phase has to be done while (or after) using and interacting with the system. This is the interactional aspect and corresponds to *End User Programmming.*

End user programming can be seen as offering alternatives to extensive programming for building applications. The question is how the end user can transfer to the system its relevant knowledge so that it can accomplish the required task. In [9], the mobile phone behavior is contextually specialized using feed forward rules. A specific interface allows to easily create new rules based on predefined concepts, brought by the programmer. The end user can specify how those concepts should be used to fulfill his needs. The evaluations showed that the end user appreciates to be able to enter its own knowledge in the system as long as the rules are not too complicated to write. The main problem is that Nokia’s system is *bringing its own predefined concepts* which might not correspond to the user’s concepts and interpretations.

In [5], the user is doing its task in the environment while the system is recording all the sensors. Using a dedicated human machine interface, the end-user can then annotate the recorded data and a context model is constructed using machine learning approaches. This annotation phase is an interesting way to transfer knowledge and concepts from the user to the system while letting the system building its own internal representation. The main drawback is that it can be rapidly painful and sensor data annotation is done offline (not during the interaction). Ganneau and al. [7] are directly learning (using bayesian networks) the situations' description without a post-annotation phase. The interaction for knowledge injection is more transparent for the end user.

On-line transparent interaction is more natural and corresponds to the interactional aspect of Dourish but can be ambiguous (see section 3). Off-line specification using a dedicated interface can allow the end user to specify more precisely but is painful. We propose a global approach based on both approaches: transparent user interaction and dedicated off-line interactions to disambiguate the injected knowledge.

3 Situated End User Programming

In an idealist scenario people should teach machines through a social and collaborative process. Learning to act appropriately in social situation is a first step toward a better interaction. In a previous work [1] we have sought to train an association between behavior and social situation. However the success of this approach relies on getting both human and system to share a common understanding of social situation.

3.1 Mutual Intelligibility

Salembier et al. [11] review research that point out the importance of the mutual access to contextual information in collaborative work. The better the mutual understanding the greater the collaboration. This mutual understanding is not necessarily achieved by sharing mutual knowledge but rather by the concept of mutual manifestness. Introduced by Sperber and Wilson this concept is weaker but empirically more adequate than the theory of mutual knowledge which has the characteristic to produce regression at infinity.

For Sperber and Wilson "a fact is manifest to an individual at a given time, if and only if, this individual is able at this time to represent this fact mentally and to accept his representation as being true or probably true" [12]. Following this notion, Sperber and Wilson define the one of cognitive environment. A personal cognitive environment (PCA) is defined as whole facts which are manifest for a given individual. A shared cognitive environment (SCE) indicates all the facts which are manifest to several individuals. This simply means that they are able to perceive or deduce the same facts, and not that they share a belief, a knowledge, or a representation concerning those facts. The mutual cognitive environment (MCE) indicates a shared cognitive environment in which the identity of individuals who have access to this environment is manifest. As they share the same environment, they can establish an interaction in relation to their common perception of contextual events. Salembier points out that the notion of cognitive environment does not take into account the activity of individual. He [11] then proposes the definition of shared context, which reduce mutual cognitive environment through activity filtering. Thus shared context is a set of contextual information or events mutually manifest for a set of actors, at a given time in a certain situation, taking into account their perception and cognitive abilities, their task, and current activity.

Considering social learning as a collaborative process and context as a key issue [2] in interaction between human and computer, we argue that mutual intelligibility is the key toward a social human-computer interaction.

3.2 Situation Models

In the previous section, we emphasize the importance of mutual intelligibility in social learning and particularly the needs for a MCE between both human and system. However it is not clear what defined the facts that compose cognitive environment. Johnson-Laird in [8] introduced situation model as cognitive theory for human mental models. A situation model is a mental representation of a described or experienced situation in a real or imaginary world [13]. It is commonly defined as consisting of entities and of relations between those entities. While this model, as well as much of the subsequent literature in this area, has been concerned with spatial reasoning or linguistic understanding, these concepts can be adopted for the construction of software systems and services for understanding social interaction. In [3], we describe the use of situation models for observing and understanding activity in order to provide context aware services. Situations are defined as a set of relations (predicate) between entities (agents, objects or abstract concepts) where entities are sets of properties.

Using this formalism we define a fact as an entity or a relation that can be either observed or deduced by an agent. A cognitive environment is then defined by the set of relations between entities that are manifest by this agent. Thus the shared context between agents (human or machine) is defined by the intersection of their situation models taking into account their perception and cognitive abilities, their task, and current activity. A fact (entity or relation) has the property to exist and is associated to an abstract representation that we call concept. Two facts can be instance of the same concept. Concept is for fact what a class is for object. Observed facts are any instance of concept that can be observed and represented as an entity or relation. Such entities include time of the day, weather, agent, object. Observed relation include spatial relation. Inferred facts are any instance of concept that can be inferred from other facts (observed or inferred). Inferred entities can include group, agent's activity, agent's task. These facts are not directly observable properties but rather interpretation.

3.3 Socially Guided Environment

The concept of mutual manifestness as part of the relevance theory has been concerned with human communication. In the previous section, we argued for the need of such an ability in human-machine interaction. Considering the main differences in term of cognition and perception between human and computer, this theory gives us a clear understanding of why perceiving social context from a human perspective is a difficult problem. Although this let us to propose our approach of end-user programming. In Relevance Theory [12] Sperber and Wilson present communication as the process of making certain facts more manifest than others. Thus in our approach both agents are provided with tools allowing them to alter each other's cognitive environment. Because of the fundamental difference between human an machine, we need to introduce the following condition. A fact is manifest for the system if and only if the related concept is also manifest. Then by definition a concept is manifest if and only if it belongs to the cognitive environment. A concept is then mutually manifest if and only if it has been acknowledged by the user. Therefore, the first step toward making a fact mutually manifest is to teach the system new concepts.

The first tools provides the user with a method to teach the system whatever perceptual concept he thinks is relevant. This concept is tokenized from the environment using bounding volumes such as blob or frustum constructed by user hand gestures. This tool is well adapted to make manifest spatial concept such as a chair, a desk or an area. At the same time, the system is able to display different views of its cognitive environment by displaying the relevant facts it is able to perceive or deduce from the environment. This second tools presents two benefits: first it provides transparency to the user, then it allows the user to respond whenever he did not agree. One of this view is provided by an interactive table. The table provides a window through the system's mind which lets the user browse and interact with the contained concepts and facts. Different actions are then proposed to the user such as teaching a new abstract concept, adding tags to a already learned concept, adding or removing a concept from the mutual cognitive environment. Tags are a simple way for the user to give interpretation to newly extracted concepts (eg. Chair, Desk, Table). The user can also see the current and previous system's situation models and related action, thus enabling the user to tell the system the action it should perform in a given situation. The interface provide also a tools for the correct assignment of reward in the case of social reinforcement learning [1]. Finally we integrated an attentional model to control the manifestness of a fact. The greater the attention of the user toward a fact the more manifest the fact. That way entity closest to the user for example will be more manifest than entity outside the environment.

4 Experimental Evaluation

In this section, we demonstrate our preliminary results with a step by step case study which illustrates a standard scenario in a context aware environment. The goal is to show how a user and a system can build conjointly a mutual cognitive environment.

4.1 Experimental Setup

The experiment takes place in the PRIMA SmartRoom, an environment specifically designed to observe (using cameras and microphones) and interact with users as they go through normal day-life activity such as work, meeting, rest, etc. The system is capable of tracking the position of persons as they evolve through the environment and also provides a rough estimate of the personâĂŹs posture. The system is also equipped with a steerable projector with automatic image rectification able, given a specific region, to transform a planar surfaces in a projection area.

4.2 Case Study

One scenario that comes to mind when working with smart environment is the meeting scenario. In such scenario, smart environments could provide different services such as projecting content of the meeting at the correct place. In the following, we will illustrate the different steps that lead to the construction of a mutual cognitive environment intended to provide our smart meeting system with the needed concepts to characterize/recognize the relevant situations.

This construction process is organized in two stages. The first step is a usual top down reflection where the user builds a mental representation of the situation he wants to explain by selecting relevant assumptions. A possible situation model could be a "**lecturer** *is close to* an **audience**". Then a **lecturer** *is a* "**person** *close to* a **projection area**" and so on. The second step is an iterative process where both human and system interact to make relevant information mutually manifest. Using different modalities the system acknowledges the user of the different facts and related concepts that are manifest in the environment. The user on its side, introduces the system with missing concept. For this experiment the environment already knows some basic concepts, provided by software components, such as the entity-concept **person**, **audience**, **region** and the relational-concept *close* to and *sit on.* The following concepts are "mutually manifest" however **chair** and **lecturer** are not. We could only expect the environment to interpret a correct situation model once **chair** and **lecturer** will be mutually manifest. Then, the steerable projector will project accordingly the content of the meeting at the correct place in the environment.

When the user enters the room, the only facts belonging to the MCE is the user itself. The system is able to perceive the user position and the user knows it, thus this fact is mutually manifest. The first action performed by the user in the environment is to defined the concept of **projection** area which is then integrated to the MCE. This step is done by performing a hand gesture surrounding this area and tagging the extracted region on the interactive table as **projection area)**. Then the user pursues by extracting the different sitting areas (see Figure 1) to which he will associate the tag **chair**.

This far, the system is able to observe new facts from its environment. For example any person sitting on the extracted chair or sofa will now be manifest to the system as the related concepts, **person**, **chair** and *sit on* are known. The last missing concept is the one of **lecturer** which is a deduced-like concept. In order to do that, the user interacts with an interface displayed on an interactive table. The interface shows the different concepts mutually manifest as well as observed and deduced facts. Abstract concepts are defined from other set of concepts

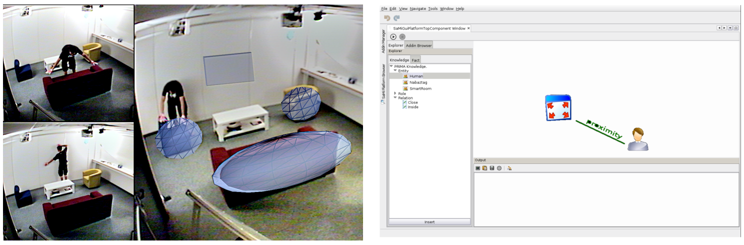

Fig. 1. Teaching concepts (left) and learning abstract concept (right).

in relation, thus user has to construct a diagram relating sub-concepts together. This operation is achieved by different drag&drop operation. In our example, the diagram containing a **person** *close to* a **projection area** is labeled as **lecturer** (see right part of Figure 1).

During the real meeting presentation (see Figure 2) the user can, at any time, teach the system news action according to its interpretation of the current situation. The right part of Figure 2 presents the current interpretation of the situation by the system (remark, relation are not present). System can be taught to trigger the steerable projector service, and so the topic of the meeting will be displayed onto the correct projection area.

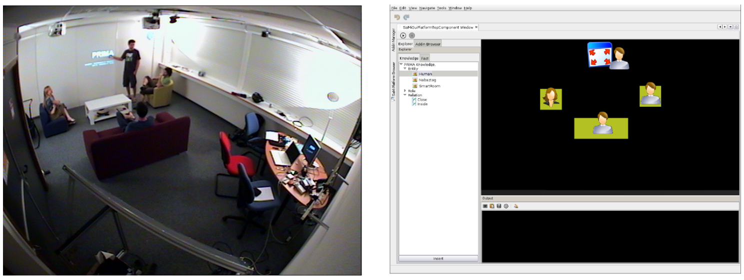

Fig. 2. Presentation meeting (left) and interpretation of the presentation meeting by the system.

5 Conclusions

In an attempt to design context aware systems able to acquire social common sense from day to day interaction with non-expert user, a key challenge is to provide systems with methods to perceive social situation from a human perspective. We cannot expect developer to handle all possible social situations a system can encounter in its working life. An alternative inspired by end user programing offers better perspectives.

Our proposed approach provides transparent interaction and learning tools to build a mutual cognitive environment between the user and the system. We argue that mutual cognitive environment plays a significant role in the construction of situation models for social situation. Because system and user do not have the same perceptive and cognitive capabilities we propose a method where user will interactively (when the needs comes) teach to the system the concept that he thinks is relevant. Concepts are learnt by segmenting the environment using simple hand gestures and using an interactive table as a cognitive window through the system mind. The preliminary results are very encouraging and validate our approach.

References

1. R. Barraquand and J. L. Crowley. Learning polite behavior with situation models. In *HRI '08: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 209–216, New York, NY, USA, 2008. ACM.
2. J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan. Context is key. *Commun. ACM*, 48(3):49–53, 2005.
3. J. L. Crowley, P. Reignier, and R. Barraquand. Situation models: A tool for observing and understanding activity. In *Workshop People Detection and Tracking, held in IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.
4. A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, 2001.
5. A. K. Dey, R. Hamid, C. Beckmann, I. Li, and D. Hsu. a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40, Vienna, Austria, 2004. ACM.
6. P. Dourish. What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8(1):19–30, February 2004.
7. V. Ganneau, G. Calvary, and R. Demumieux. Learning key contexts of use in the wild for driving plastic user interfaces engineering. In *Proceedings of the 2nd Conference on Human-Centered Software Engineering and 7th International Workshop on Task Models and Diagrams*, pages 271–278, Pisa, Italy, 2008. Springer-Verlag.
8. P. N. Johnson-Laird. *Mental models: towards a cognitive science of language, inference, and consciousness.* Harvard University Press, Cambridge, MA, USA, 1983.
9. P. P. Korpip, E. Malm, T. Rantakokko, V. Kyllonen, J. Kela, J. Mantyjarvi, J. Hakkil, and I. Kansala. Customizing user interaction in smart phones. *Pervasive Computing, IEEE*, 5(3):82–90, 2006.
10. B. Reeves and C. Nass. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge, [1st] edition, 1996.
11. P. Salembier and M. Zouinar. Intelligibilité mutuelle et contexte partagé. Inspirations théoriques et réductions technologiques . *@CTIVITES*, 1(2), 2004.
12. D. Sperber and D. Wilson. *Relevance: Communication and Cognition.* Blackwell Publishers, December 1995.
13. R. A. Zwaan and G. A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123:162–185, 1998.