# Theory and Practice of Social Reasoning

# Experiences with the iCat

**Frank Dignum, Bas Steunebrink, Nieske Vergunst, Christian Mol, John-Jules Meyer**[1]

**Abstract.** In this paper we discuss how a social companion robot can be programmed using agent technology. An important challenge is to combine the physical and conceptual elements of social relationships.

## 1 INTRODUCTION

The term "social reasoning" can be used in many different ways and in many different settings. One can argue that if one is planning a holiday with the family, social reasoning is needed to find a destination that is, in some sense, socially optimal. For this kind of reasoning one could use game theory or something related. Our setting, however, is of a different nature. We aim to program a sociable robot on the iCat, developed by Philips. The iCat is a static robot (no legs or wheels) that can only move his head and create facial expressions. Some pictures of the expressions can be seen in figure 1.

**Figure 1.** iCat showing emotions

The iCat can see through a camera in its nose and hear through microphones in its paws. It also has a proximity sensor in his left paw and touch sensors in its paws and ears. Besides the facial expressions the iCat can also talk or send information through a network link to other devices (in this way it can e.g. switch on a tv or a lamp).

The iCat platform comes with OPPR which allows you to create animations in a simple way using timelines and/or LUA scripts. OPPR also takes care of lip sync for speech output and of the blending of different outputs such as when the iCat has to simultaneously smile and talk.

In our project we are using the iCat as a kitchen companion that can assist persons in the kitchen with cooking, making shopping lists, etc. In this setting we need social reasoning in order to find e.g. a recipe that suits the whole family, but we also need social reasoning over the physical, real-time interaction. For instance, the iCat can give a complete recipe in one time, but probably the person needs to have instructions spread out during the cooking period (at the time they are needed for the next step in the preparation). And even on a lower level the iCat should check whether the person pays attention or is getting bored (by means of e.g. gaze tracking) with a conversation in order to switch to another mode.

We assume that for a high level of social behaviour the robot should be conscious of goals, desires, values, preferences, etc. of itself and the person and also how these relate (e.g. common goals, common beliefs, contradictory goals and beliefs). This led us to the use of the agent paradigm as a means to model and implement the reasoning on the iCat. Agents also are defined in terms of goals, intentions, plans and beliefs and thus seem a very good fit.

However, the traditional agent architecture can best be depicted as in figure 2. This follows the sense-reason-act loop. In such a loop the agent senses the environment, updates its beliefs and based on its current goals and intentions checks which action it should perform next.

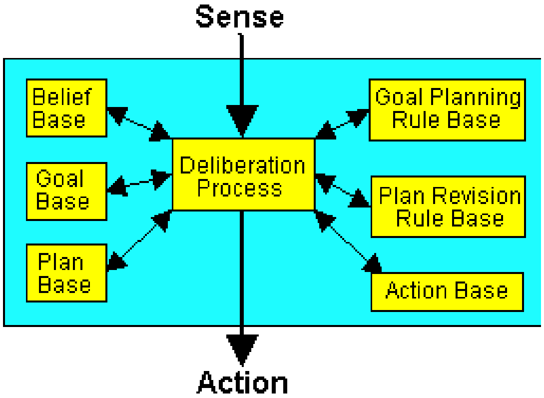

**Figure 2.** architecture of BDI agents

In this architecture the reasoning part is autonomous and cannot be interrupted. It is a well-known fact that this feature can lead to big problems in real-time environments where quick reactions to changing situations are necessary.

The most well-known social robots Max and Kismet [1,3] have thus been implemented mainly using a reactive architecture in which the robot directly responds based on the inputs it gets from the environment (although in Max [3] also goals are used and in Kismet [1] "urges" play a role).

Since real-time aspects are very important for social behaviour and thus for social robotics we are trying to combine

---

[1] Dept. of Information and Computing Science, Utrecht University, The Netherlands. Email: {dignum,bass,nieske,Christian,jj}`@cs.uu.nl`.

the traditional BDI architecture with the more reactive approaches that already exist. In the next section we sketch a short overview of the architecture and will explain shortly which role the social reasoning takes here and how this is modelled in this architecture.

## 2 ARCHITECTURE

In figure 3 we show the architecture that is used to model the social robot as we are implementing it on the iCat.

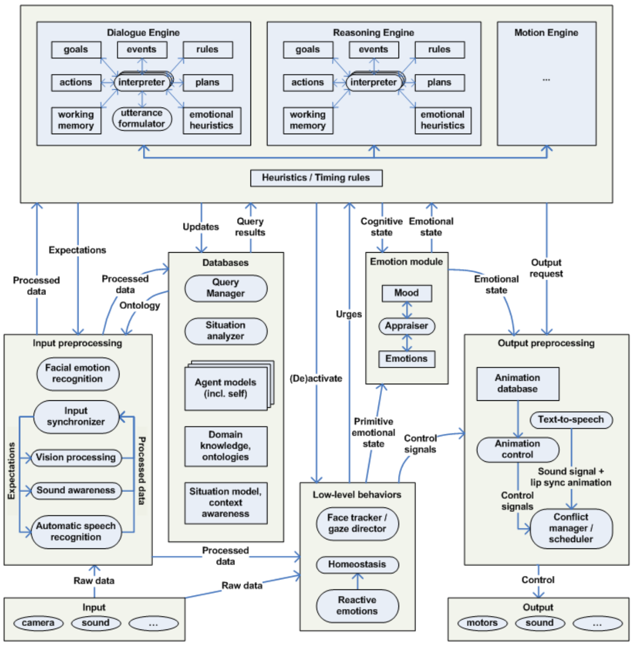

**Figure 3.** architecture of a social robot

Although we will not discuss the architecture in detail (we have another paper in preparation for that [4]) the figure illustrates nicely that the architecture of a social robot is a lot more complex than that of a traditional agent. This complexity stems from the combination of the following two points:

1. Social behaviour encompasses many different elements that should be combined.
2. Real-time aspects are important for social behaviour.

As an example of point one, a social robot has to interpret and give social cues. In order to interpret a social cue the robot should be aware of the emotions of the user, have a user profile and history, have domain knowledge and be aware how its position is in both the physical and social relation with the user. If one takes only the first point into account, one could still use a traditional agent architecture. This architecture could have a complex belief base encompassing models of the user(s), domain knowledge, history, etc. However, the second point indicates that there is no unlimited time to deliberate on its actions. For many groups in robotics, the real time aspect prompted them to start of with a reactive system that could respond in time to an event. Of course this was one of the reasons for Brooks to propose a layered architecture in which each layer builds on a (reactive) layer below [2].

The way we try to solve the problem of combining a very rich deliberation process with real-time aspects is by splitting up the different elements of the social reasoning in separate components that can run in parallel but interact through a few main deliberation loops. On the highest level we distinguish three deliberations. One for maintaining the dialogue with a user, one for general goal achievement and one for motor behaviour. The last one is just drawn for completeness as it is not needed for the iCat which only can move (some parts of) its head.

These deliberation cycles are connected with lower level processes on both the input and output side. Besides these two traditional parts we also formed this into a kind of hybrid Brooks architecture and have a low-level reactive module that takes care of primary reactions and urges such as hunger, sleep, etc.

The main challenge in using this architecture is not so much the deliberation processes on the highest level, but determining which data to feed into these high level deliberation processes and what to handle in the other modules. E.g. it is clear that one wants some geometric reasoning to take place on the vision input. This can be used to reason that a ball is moving. However, does one just indicate that the ball is moving or also the change in position or speed? Depending on what the deliberation loop is doing different types of data might be important at different intervals. If the robot is contemplating to intercept the ball, it probably wants to know as much as possible about the position and speed changes of the ball. But if it is actually talking with the user about dinner, the ball is only a distraction and no data is needed except that the ball is moving.

Our thesis is that social reasoning depends for a large part in determining how to zoom in and zoom out of the different processes that comprise the social situation and combine the results of these processes in a coherent way. This can be (partly) achieved by using things like “focus of attention” and maintaining a “shared situation awareness”. Both terms are however not very precisely definable and need more exploration to be truly useful.

## 7 CONCLUSIONS

We have sketched an architecture for social robots and tried to indicate the position of social reasoning from the perspective of this architecture. The work is an ongoing effort and we need more experience with an implementation to be able to sustain our claim that this architecture helps in modelling and implementing social robots. An important aspect for us is the question how to actually program a robot like this. Which parts of the architecture should be domain independent, which programming language should be used for the different modules and how should they be connected.

## REFERENCES

[1] C. Breazeal. Designing Sociable Robots. MIT press (2002).
[2] R. Brooks. Intelligence without representation. *Artificial Intelligence* 47, pp. 139-159, (1991)
[3] S. Kopp, L. Gesellensetter, N. Krämer, I. Wachsmuth . A conversational agent as museum guide -- design and evaluation of a real-world application. Panayiotopoulos et al. (eds.): *Intelligent Virtual Agents*, LNAI 3661, pp. 329-343, Springer-Verlag, (2005).
[4] B. Steunebrink et.al. A generic Architecture for a Companion Robot. (submitted to MARS 2008)