AGENT FACTORY: TOWARDS SOCIAL ROBOTS

G.M.P. O’Hare, B.R. Duffy, R.W Collier, C.F.B. Rooney, R.P.S. O’Donoghue

PRISM Laboratory, Dept. of Computer Science, University College Dublin (UCD), Belfield, Dublin 4, Ireland
{Gregory.OHare, Brian.Duffy, Rem.Collier, Colm.Rooney, Ruadhan.ODonoghue}@ucd.ie

Abstract.
This paper advocates the application of multi-agent techniques in the realisation of social robotic behaviour. We present the Social Robot Architecture, which integrates the key elements of agent-hood and robotics in a coherent and systematic manner. This architecture seamlessly integrates, real world robots, multi-agent development tools, and VRML visualisation tools into a coherent whole. Using these elements, we deliver a development environment, which facilitates rapid prototyping of social robot communities.

Keywords: multi-agent systems, agent architectures, agent-based robotics, co-ordination and collaboration, virtual reality.

1 Introduction
The primary concern of this paper is that of Social Robotics. Our research deals with the architecture whereby robot communities can engage in opportunistic collaborative behaviours in the solution of shared tasks.

This paper demonstrates the application of multi-agent techniques in the realisation of social behaviour between a group of autonomous mobile robots. To this end we commission Agent Factory, a software development environment that facilitates the rapid prototyping of Multi-Agent Systems (MAS). Agent Factory offers a conduit through which robot control can be governed by a deliberative agent architecture, specifically that of a Belief Desire and Intention (BDI) architecture. In addition, Agent Factory supports not only the creation of the social robotic community but also the subsequent experimentation and visualisation of their behaviour.

We integrate the key elements of agent-hood and mobile robotics in a coherent and systematic manner leading to the development and implementation of the Social Robot Architecture. This architecture seamlessly integrates real world robots, multi-agent development tools, and virtual reality visualisation mediums into a coherent whole. We therefore deliver a development environment, which facilitates rapid prototyping of social robot communities.

Section 2 offers an introduction to BDI architectures. Section 3 provides an overview of Agent Factory. An introduction to agent-based robotics is presented in section 4. Section 5 describes the Social Robot Architecture with section 6 animating its use. Discussion and conclusions are presented in section 7.

2 Belief Desire Intention (BDI)
Much research work has been commissioned on Multi-Agent Systems (MAS) and Distributed Artificial Intelligence (DAI) [BG88], [DLC89], [OJ95]. Specifically, competing agent architectures have been proposed in the literature. Two major architectural schools have emerged, namely those of the reactive system school and the deliberative system school. The former has predominated in the arena of autonomous mobile robot control. In this paper we advocate the synthesis of reactive and deliberative reasoning. In the delivery of computationally tractable models of deliberative reasoning, one approach that has gained wide acceptance is to represent the properties of an agent using mental attitudes such as belief, desire, and intention. In this terminology, an agent can be identified as having: a set of beliefs about its environment and about itself; a set of desires which
are computational states which it wants to maintain, and a set of intentions which are computational states which the agent is trying to achieve. Multi-agent architectures that are based on these concepts are referred to as BDI-architectures (Belief-Desire-Intention) [RG91] [Jen93] [OJ96], and have recently been the subject of much theoretical research. Proponents of the BDI approach argue that the understanding of the dynamics of these mental attitudes and their intimate interdependencies, is crucial in achieving rational agent behaviour.

3 Agent Factory

3.1 What is Agent Factory?
In essence Agent Factory is a distributed environment for the rapid prototyping of intelligent agents. More complete descriptions of Agent Factory are presented elsewhere in the literature [OA95] [Col96] [OJ96] [OCC+98]. Agent Factory has been specified using the Vienna Development Method and realised using ObjectShare’s implementation of Smalltalk-80, the VisualWorks integrated development environment.

Agent Factory is a member of the class of systems that embraces the BDI philosophy. The system offers an integrated toolset that supports the developer in the instantiation of generic agent structures that are subsequently utilised by a pre-packaged agent interpreter that delivers the BDI machinery. Other system tools support interface customisation and agent community visualisation. In creating an agent community three system components must be interwoven, those of agents, a world and a scheduler. The next section describes the high level architecture.

3.2 Schematic Functional Architecture
The development environment is based around Component Design Hierarchies (CDH) that are extensions of the standard Object Hierarchies in the OOP Paradigm. The development of specific agent communities requires the instantiation of particular designs from these design hierarchies. Three particular types of components have been identified as necessary for the development of agent communities: agents, schedulers, and world interfaces. Correspondingly, there is a CDH for each type. See figure 1.

The agent is the main computational unit of Agent Factory, it combines a series of attributes that represent and support the Mental State model of the agent, and a set of methods (the actuators). Agents are executed using an Agent Interpreter. This interpreter currently controls agent perception, and commitment management which considers the adoption, revision and realisation of Commitments.

The scheduler controls execution of the community, using an algorithm that exploits parallelism where possible.

Finally, the world interface acts as a medium between the problem domain, the community it is being developed for, and the other components of the Agent Factory system, as seen in figure 1.

Figure 1: Agent Factory
more recently the area of multiple robots has demanded considerable attention. There are numerous advantages in the use of multiple robots, these include *inter alia* distributed capabilities; parallelism; task and load distribution; increased functionality with minimal complexity. However, achieving coherent behaviour presents considerable challenges, none more acute than overcoming problems of co-ordination and interference. It seems clear that multi-agent techniques are amenable to transference to systems of multiple autonomous robots, in particular to addressing the problems of co-ordination and interference.

Initial work on agent based robotics emerged from cellular robotics where the robots had limited functionality and relied on swarm like intelligence to achieve their desired task, typically exhibiting *emergent* capabilities. Fukuda *et al* conducted research in the early stages of agent robots on both multi-robots in DARS (Distributed Autonomous Robotic System) [CF95] and reconfigurable robots within CEBOT (Cellular Robotic System) [Fuk+89].

Dudek *et al* in [DJM96] presents a taxonomy that classifies multi-agent systems according to communication, computational and other capabilities. Arkin and Balch [AB98] developed their reactive strategy for multi-agent co-operation with robots searching for trash objects, which they grasp and carry to wastebaskets. This research is concerned with the development of behaviours for formation maintenance in heterogeneous societies of mobile robots.

The RoboCup initiative [Rob95] provides a standard controlled environment for a team of multiple fast-moving robots to perform tasks, namely soccer playing, under dynamic real-time circumstances.

"The Robot World Cup Initiative (RoboCup) is an attempt to foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined"

The various technologies being researched include design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion. RoboCup also provides a *softer* transition from theory to reality through its standard environments.

Balch *et al* researches the impact of communication in reactive multiagent robotic systems in [BA94]. Balch later uses the soccer scenario for evaluating agents in terms of performance, policy convergence, and behavioural diversity [Bal97].

5 Social Robots Architecture

The motivation behind this research is to demonstrate, in a tangible form, the correspondence between systems of multiple agents and systems of multiple robots.

5.1 Social Robots

We introduce the term *social robots* here. Pre-existing research has been undertaken, most notably by groups at the Universities of Edinburgh and Reading, which interprets sociality from the perspective of communication, learning and human machine interaction. Researchers at the University of Essex, under the auspices of the EOS+ [EOS+] project use the techniques of Distributed Artificial Intelligence and artificial societies to study the emergence of human social complexity.

It is our conjecture that a distinction exists between *societal robotics and social robotics*. The former represents the integration of robotic entities into the human environment or society, while the latter deals specifically with the social empowerment of robots permitting opportunistic goal solution with fellow agents. This paper offers valuable insights into the delivery of, and experimentation with social robots.

5.2 Social Robot Architecture

The computational machinery needed to facilitate team building and collaborative behaviour is non-trivial.

We describe the *Social Robot Architecture*, which goes some way toward achieving this through the judicious synthesis of the reactive model with that of the deliberative model.

The layered architecture (figure 2) has three fundamental elements: the deliberative level provided via Agent Factory, the robot level, and the environment level, which we consider in more detail in the subsequent sections. The control architecture offers basic reactive behaviours for reflex robot responses to unexpected or dangerous events. These constitute a set of primary *survival behaviours* for the robot. Goal oriented behaviours are delivered through the intentional agent structures of Agent Factory. Shen [SAC96] investigated a similar layered behaviours approach.
Agent Factory supports inter-agent interaction (figure 3) enabling the resource bounded robot to usurp minimal computation on such activities, thus maximising resources for perception, reasoning, planning and action. Any process intensive action (i.e. planning) retards the real-time reflexive nature of the agent. Typically, more deliberative behaviour necessitates increased high level inter-robot communication, planning and deduction which could result in a loss of perception granularity.

5.2.2 The Robot Level

A series of fundamental behaviours are implemented at this level. They provide the basic *survival kit* of the robot necessary for such dynamic unpredictable environments. These *Survival Behaviours* include such behaviours as *avoid_obstacle*, *stop*, and *retreat*.

The sensory information received from the ultrasound and bumper sensor rings is processed at this level, resulting in clear agent-events being generated and communicated to Agent Factory.

5.2.3 The Deliberative Level: Agent Factory

The deliberative layer provides the deliberative machinery. This is achieved through the BDI architecture described in section 3. Beliefs are generated based on its belief set, and are updated with the receipt of new agent states or events from each robot. Our agents interact via an Agent Communication Language (ACL) based upon Speech Act Theory [Aus69]. The language, *Teanga*, is dealt with in more detail in section 5.3.

*Agent events* form the basic catalysts for robot communication and result in information being sent to the Agent Factory high-level controllers (agents). Raw sensory data could be sent straight to Agent Factory however, this requires further processing to firstly filter and thus achieve attention focusing and to deliver the added value needed by our agents. Rather than burden Agent Factory with this low

---

1 Aengus, Bodan and Bui are ancient characters in Irish legend and their graves can be found at Bru na Boinne, Ireland.
2 *Teanga* is Irish for “language” and interestingly is an anagram for a agent.
level processing, this is undertaken at the robot level, distilling the data down to a more useful form.

A scheduler in Agent Factory allots time slices to individual agents. As an agent enters its allotted time slice, it starts by gathering perceptions. Local robot processing filters these, generating a set of agent events that have occurred since the preceding time slice. The perception process within Agent Factory deals with converting these perceptions into beliefs and adding them to the belief set providing the agent with an up to date model of its current perceived situation. This situation is then analysed and the agents’ commitments are updated. Those commitments that the agent made for realisation in the current time slice are analysed and, where possible, agent_commands are dispatched to the robot.

By way of example a team of robots may have a goal of mapping their environment. Decomposing this problem further, the individual robots might have to map constituent rooms, which in turn are made up of walls and a door. By analysing the data gathered from the sonar’s these components might be identified. If a wall is found the robot then generates an agent event of the form, wall_from(X,Y,Xi,Yi). Upon receipt the robot agent will revise it's belief set and accordingly add the belief Bel(wall_from(X,Y,Xi,Yi)).

5.3 Teánga

Our agents interact via an Agent Communication Language (ACL), called Teánga, which is based upon Speech Act Theory [Aus69]. Teánga consists of 4 basic categories of communicative acts (messages): informatives, directives, commissives and declarations. This categorisation was developed from a classification of performatives proposed by [Searle76]. Within each category there are more specialised types of communicative acts, e.g. drop_commitment. The language is designed as a carrier for an application-dependent content language. We do, however, place some constraints on prospective content languages. In the context of the Social Robot Architecture the content language must be able to allow, amongst other things, the representation of actions (including speech acts) and their status e.g. done, doing; and an agent’s mental states, e.g. beliefs and commitments.

One of the main reasons why we chose to develop our own ACL rather than work with an established language such as KQML [LF94] or FIPA’s ACL [FIPA97] is that our language should be compositional. We wish to be able to support nested speech acts (and speech acts contained inside composite actions, e.g. plans). To implement a nested communicative act in KQML requires that the content be a KQML message. However, in KQML the content language is independent and there is therefore no content checking. “A disadvantage of content independence is that it prevents the content from being checked for compatibility with the speech act type” [CL95].

The communicative acts within the real world are sent via wireless Ethernet.

5.4 Reality and Virtuality

One of the key tenants of our research has been the provision of multiple views of the operation of the same robot system. Consequently, the environment level provides two system views. The primary view provides a physical perspective of the Nomad Scout II’s navigating the robot world. We supplement this with an abstract view to support behaviour generation and testing. The secondary view is a virtual reality perspective provided via the Agent Factory Visualiser which delivers a 3-D VRML world via the Internet (figure 4).

![Virtual reality view of the IMPACT research room](figure 4)

Herein we harness the advantages of using virtual environments, by directly relating virtuality and reality in a seamless manner. Such alternate views permit multiple views, information hiding and abstraction, system interaction, and behaviour scrutiny via snapshots and recordings.

Our architecture supports the specification and invocation of robot experiments via a Java Internet interface. Each world is activated and their views synchronised.
5.5 Incremental functionality

In order to implement the complex notion of social robotics, we have broken the levels of behavioural complexity into incremental functionality as shown in figure 5.

As robot control stems from simple move and turn commands to complex notions of social robotics, this allows for systematic task decomposition and development leading to individual robot behaviours and extending to multiple robot behaviours.

A library of robot behaviours exists which adopts a subsumption model [Bro86]. Higher layers provide increasing complexity and subsume lower level functionality (see figure 5). Reactive or reflex survival behaviours are implemented at the robot level with more complex behaviours defined within Agent Factory. Each robot therefore has a degree of autonomy, which is independent of any inter-robot communication, and important in the realisation of robust complex robot behaviours.

![Figure 5: Layered social behaviours](image)

5.5 The Virtual Reality Workbench

Detailed programmes of robot tests have been performed and are characterised in the subsequent section. As a part of this research however we also wished to provide a shared global resource available and usable across the internet which supported distant robot experimentation.

This facility, the Virtual Robotic Workbench, provides a medium through which researchers can design robot experiments remotely and without recourse to the purchase of expensive robotic entities in the first instance.

Researchers can articulate their experiments across a Java interface whereby they tune certain key workbench parameters, namely:

- The Customisation of the Environment
  - The Number of Robots;
  - The world within which they are to be situated;
- The Customisation of the Robots
  - Their Name;
  - The visualisation avatar associated with each robot;
  - Mapping virtual robots to physical robots;
  - The behaviours ascribed to a given robot;
  - Their Initial Location within the selected world;
- The Task
  - The selection of the shared goal;

Figure 6 depicts the nature of the Java interface by which these are specified Once the experiment has been crafted the subsequent activation results in the observable behaviour of the robots across any suitably configured web browser. The behaviour of the robots is governed by the social robotic architecture with the higher level functions directly furnished through Agent Factory. As such the virtual experimental results and the navigation and behaviour of the robotic avatars can be seen by utilising the Agent Factory Visualiser. Detailed treatment of the Agent Factory Visualiser is presented elsewhere in the literature [OCC+98]. Agent Factory generated events governing agent behaviour are dispatched simultaneously to both the robot controller, which activates the motors on the individual physical robots, and a proxy server, which, via the External Authoring Interface (EAI) of VRML, updates the world view accordingly, resulting in the movement of the virtual robot. The difficulty is in trying to achieve synchronisation between these two views3. Figure 7 illustrates the Agent Factory Visualiser and gives the reader a feel for the potential which the virtual robotic workbench offers.

---

3 Clearly only in certain limited circumstances will it be possible for a corresponding physical experiment to be conducted at the same time. It is possible to provide a richer palette of virtual worlds for experimentation.
The benefits of the application of virtual reality to robotic experimentation are many-fold. Perhaps, the most obvious benefit is that it allows for remote behaviour observation with multiple perspectives. There is also the fact that any website that provides remote real-robot experiments is constrained by the number of physical robots available. The robots may be in use or offline. Our system can utilise simulated robots if the real-robots are unavailable thereby providing the user with an accurate 3-D visualisation of the experiment. Finally, on a more logistical note there is maximum resource utilisation through the participation of different research groups.

6 Application

With the building of a virtual reality representation of the PRISM research Laboratory and the robots visualised in this environment (as shown in figure 8), we have a testbed within which to demonstrate that a constructive link between reality and virtuality can be achieved.

This link is created and maintained through the Agent Factory World Interface (see figure 6).

This web-based interface allows the user to create, view and manipulate agents. These agents may then be situated in worlds and linked with real robots. A world can therefore exist in two environments: physically and in Virtual Reality (VR) as shown in figure 9.

Robot experiments are characterised by firstly selecting a world, subsequently situating robot(s) in this world, and finally ascribing behaviours to these robots.

6.1 The Waltz

One such experiment, which exercises the degree of architectural and behavioural functionality required to demonstrate the applicability of the social robotics architecture, is that of a waltz duet. The specific waltz is the “Box Waltz” which consists of a simple square shape repetitive motion. The objective is for the two robots, Aengus and Bui to perform the waltzing behaviour side by side in a space restricted environment.

Figure 10 shows both a real and virtual perspective of the experiment and the synchronous motion of the robots.

Figure 11 demonstrates the robot trajectories while performing the waltz, clearly showing the behaviour modification when one robot encounters an unexpected obstacle or wall while also utilising the concepts of collaboration and communication to maintain a co-ordinated behaviour.

This experiment clearly demonstrates a robust system which, through the fusion of reactive survival behaviours and high-level deliberation, communication and co-ordination, can achieve complex goals in a real-time, real-world environment.
Figure 11: The two robot trajectories of the performed behaviour in a rectangular room

7 Discussion and Conclusions

This research has introduced Social Robots and an accompanying Social Robot Architecture. The architecture provides firstly, a powerful robot control mechanism, which integrates reactive and deliberative features and secondarily, a rich visualisation medium, which offers seamless, contrasting, yet consistent views of social robotic behaviour. Subsequently social robot experiments may be observed and analysed through these mediums.

With the breakdown of robot behavioural complexity, this has enabled the realisation of the social robotics concept via a series of developing stages of complexity and functionality.

To date, experimental evidence has demonstrated that constructive and coherent collaboration is achievable in restricted task domains. We are confident that our social agent architecture is scalable and ongoing research seeks to demonstrate its applicability to different real-world problem domains.

9 Acknowledgements

We gratefully acknowledge the support of Enterprise Ireland through grant No.’s SC/96/621 COMEDY and SP/97/08 IMPACT (Initiative for Mobile Programmable Agent-based Computing Technologies).

10 References