The Social Robot Architecture: A Framework for Explicit Social Interaction

Brian R. Duffy (Brian.Duffy@ucd.ie)
Mauro Dragone (Mauro.Dragone@ucd.ie)
Gregory.M.P. O'Hare (Gregory.OHare@ucd.ie)
Department of Computer Science, University College Dublin
Belfield, Dublin 4, Ireland

Abstract

This paper details the Social Robot Architecture, a framework for explicit human-robot and robot-robot social interaction. The core mechanisms for realizing a robust robot control architecture involving a synthesis of reactive, deliberative, and social reasoning mechanisms are presented and discussed. In addition, the Virtual Robotic Workbench is briefly described which demonstrates the coherent integration of both physical and virtual social robots in the human social space. These technologies are implemented on both autonomous mobile and humanoid robots.

Introduction

The development of social robot control strategies has evolved over many generations. Walter Grey's original "Machina Speculatrix" was one of the first demonstrations of emergent reactive social behaviour. A number of surviving photos of Elmer and Elsie (ELectro MEchanical Robots, Light Sensitive) illustrate how they use lights and light sensors to interact, such as moving towards each other, and engaging in a dance, as described in "The Living Brain" (Grey, 1953). During the last half century, differing arguments have been presented in support of strong representational approaches (Classical AI), purely reactive behaviour-based mechanisms, and the synthesis of the two. In recent years with the advent of the Social Robot and explicit strategies for inter-robot and human-robot communication, researchers have been called upon to address the complexities of social embodiment (Duffy, 2004).

There are generally two camps of social robot research. The first involves the development of explicit control architectures for (ideally) heterogeneous robots with the capacity to communicate, coordinate, and engage in complex social behaviours. The social interaction design strategy is primarily based on a bottom-up approach where the social capabilities are often additional mechanisms implemented on robotic devices. There are numerous *levels* of sophistication from simple message passing through, for example, infra-red transmitters and receivers (Dautenhahn, 1995) to sophisticated Belief-Desire-Intention-based control paradigms (Rao and Georgeff, 1995).

The second approach to social robot research is a more top down strategy where the objective is to empower a robot with the social functionality required to engage human participants in some form of social engagement. Such systems often involve building robotic devices with some degree of anthropomorphic representation such as a head, body, facial expressions, and hand gestures (see Duffy (2003) for a discussion on anthropomorphism in robotics). The control strategies generally employ strongly human-centric interaction modalities such as speech and even models of emotion in order to realize a social interaction that is as natural as possible. While the original motivations for both camps may be different, they are not mutually exclusive.

The *Social Robot Architecture* discussed in this paper adopts the stance of an ego-centric robot control strategy, developed to socially extend to those the robot engages with, whether other robots, humans, or indeed virtual avatars. It is intended as a more generic control strategy for robots capable of explicit social behaviour.

The following section discusses a number of hybrid architectures and sets the stage for the subsequent section where the Social Robot Architecture is detailed. These mechanisms are implemented on a number of robot platforms in both real and virtual spaces, each interacting with both other robots and human participants alike.

Social Robot Control

While considerable research has sought to address the fundamental issues of real-world robustness in autonomous mobile robotics through the development of reactive architectures, deliberative architectures and strategies employing a synthesis of the two, a coherent integration of representational and non-representational approaches remains an issue.

Connell has proposed a three level architecture known as the SSS architecture that consists of Servo, Subsumption, and Symbolic layers (1992). Numerous other examples are described in the literature; Kortenkamp et al. (1998), for example, provides a good overview of many related research endeavours.

A fundamental product of the synthesis between reactive controllers and traditional planning architectures is the behaviour-based control paradigm, viewed as the decomposition of tasks into predefined behaviour modules which can ideally be activated at any given time in response to a particular sensory stimulus. Examples include work by Brooks (1986), Logan and Sloman (1998), Arkin *et al* (1993), Oka *et al* (1997), and Mataric (1990).

While clearly not a new approach, the integration of the reactive and deliberative control methodologies with the addition of the social level for embodied physical robots acting in a social environment has not been adequately addressed. Issues arise as to how the reactive and deliberative components of such a robotic architecture can be coherently integrated without compromising system robustness. If centralised mechanisms control behaviour activation and inter-behavioural conflict, problems arise when the number of layers increases with the complexity of the task-space with associated difficulties in defining all the layers, together with their associated interrelationships and dependencies.

The Social Perspective

While the desirable attributes of hybrid architectures that combine deliberative and reactive control approaches are intuitive, few incorporate strong mechanisms for explicit social interaction.

	Embodied multiple robots	Deliberative Reasoning	Communi-cation	Social Features	Leyered Structure	Explicit Hardware Independ-ence
Social Robot Architecture (Duffy, 2000)	✓	Explicit	Formal Language	✓	✓	✓
ALLIANCE (Parker, 1998)	✓	Explicit	Limited	Limited	X	✓
Billard *et al.* (1997)	✓	Limited	Semaphore	X	X	X
MICROB projects (Drogoul and Ferber 1992)	✓	Explicit	Limited	X	✓	X
DARS [Cai et al, 1995]	Simulation	Emergent	Limited	X	X	N/A
ISR/IST research (Sequeira and Ribeiro, 1998)	Simulation to date	Emergent	X	X	X	N/A
STEAM (Tambe 1997)	Simulation	Explicit	Formal Language	X	X	N/A
Social Robot Project [Nakauchi & Simmons, 2002]	Single Robot	Explicit	Natural Language	✓	X	X
MINERVA (Thrun et al, 2000)	Single Robot	Explicit	Natural Language	✓	✓	X
Donand & Jennings (1997)	✓	Limited	X	X	X	X

Table 1: A comparison of Social Robot Architecture with some existing social robot control approaches

The table above provides a relative perspective of a number of existing architectures, chosen to be representative of the different approaches, for the control of multiple robots. Two strategies for single robot control are also included as they incorporate explicit social features for human-robot interaction. Kuniyoshi (1997) demonstrates an alternate approach through the development of social mechanisms of attention with cooperation achieved through observation and imitation without explicit communication.

The Social Robot Architecture (SRA) presented in the following section has been designed and implemented with explicit social functionality designed for robot-robot, robot-avatar, and robot-human social interaction.

The Social Robot Architecture

The *Social Robotic Architecture* (Duffy, 2000) is a design methodology based upon the combination of BDI (Belief, Desire and Intention) agents, a reactive behavioural system, and an explicit social infrastructure (figure 1).

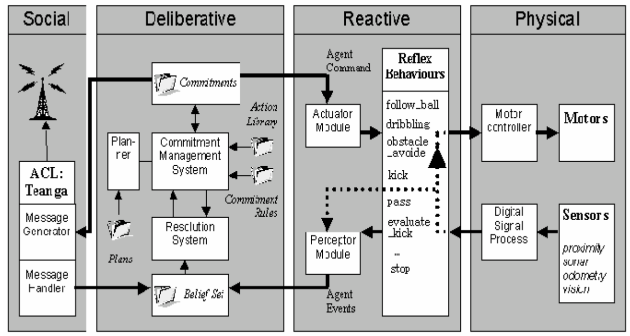

Figure 1: The Social Robot Architecture (Duffy, 2000).

Key components of the architecture are a hardware abstraction layer for heterogeneous robot platform applications, a coherent reactive-deliberative control synthesis, a Belief-Desire-Intention deliberative level developed through Agent Factory (Collier, 2001), and a FIPA compliant social level to support explicit social interaction (http://www.fipa.org). The following sections detail the individual layers of the architecture and their interaction.

In this work, we embrace the reactive-deliberative synthesis discussed earlier, employing mechanisms where expectation and attention focusing controls the accumulation of a robot's behavioural confidence through active sensing and sensor fusion techniques. The behavioural level of the Social Robotic Architecture follows guidelines emerging from early works in reactive controllers and behaviour-based control. We favour decomposition by activity rather than functionality and our behaviours modules are kept relatively simple with limited intra-communication at a local level (Brooks, 1991; Mataric, 1990).

Physical Level

The physical level provides for an ease of portability of the *Social Robot Architecture* between differing physical robot platforms. As the sensor and actuator capabilities and functionality can change, as with the corresponding digital signal processing (DSP) modules for example, a degree of abstraction from these sensor and actuator modalities is required. As this control methodology is designed to be hardware independent, the physical level is individually tailored to each hardware platform. This is achieved through the class `Body`, which provides an abstract interface to the physical layer (actuation and perception). In our work this interface is shaped so as to handle the characteristics of different robotic platforms, from generic wheeled robots (for which the command `set_velocity` is used to set translational and rotational velocities, while the function `get_position` returns the state of the odometry used to estimate the robot movements) to the dynamic properties of walking humanoids or simulated models (i.e. walking or flying avatars).

Figure 2. Some of the robots at the PRISM Research Lab in University College Dublin, which employ the Social Robot Architecture

Behavioral (Reactive) Level

The Reactive Controller component supervises the physical layer, managing a library of primitive modules named `activities` and `behaviours`. Activities are usually responsible for data acquisition (sensor drivers) and sensorial processing (e.g. feature extraction) and run in parallel within the controller. Behavioural modules implement both reflex robot responses to unexpected or dangerous events and more complex actions. These constitute a set of primary *survival skills* for the robot. Normally these modules attempt to maintain or achieve simple relations between sensorial inputs and the robot's internal state (like `avoid_obstacle`, `wander`, `follow_wall`,) or perform simple operational sequences (like `step_forward`, `turn`, `align_with_wall`).

Contrary to `activities`, only one `behaviour` among those available to the controller is used to control the body at any given time. When a `behaviour` is selected, the sensor's output is routed to that `behaviour` and the suggested commands redirected to the robot's body (see figure 2). `Behaviour` implementations do not address the specifics of what body they are controlling, thus enabling easy transfer of code from simulated to physical robots of different platforms. Only the `Body` component and the sensor drivers encapsulate the physical interactions between the robot and the environment. The standard body implementation is in fact a bridge to a native library providing access to the specific robotic platforms (e.g. through RS232).

A key characteristic of the SRA Behavioural Level is the way perception units are associated to behavioural modes. In observing that sensory-motor primitive constrains the dynamic of the interactions between the robot and its environment, this constitutes an effective motivation to structuring perception and attention focusing. In early work (Duffy et al, 2000), it has been shown how behavioural modes simplify the perceptual space and how feature detection (i.e. identifying signatures in the values returned from the sonar ring during wall following) can be used to create perception hypothesis and expectations in order to channel future structured sensing strategies, leading to the formation of perceptual evidence. This maintains the advantage of using reactive behaviours to deal with real-world complexity through using the real world itself as its own best model as, otherwise, using a form of abstraction of low-level events at the deliberative level can lead to difficulties in maintaining these advantages.

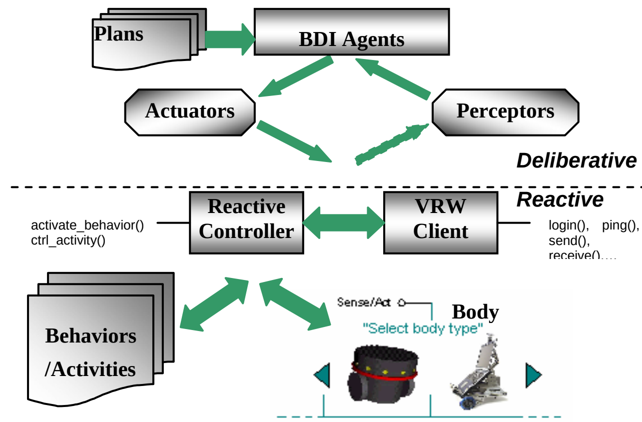

Figure 3: Structure of the Social Robot Agent.

This is managed in the Social Robot Architecture through the use of evidence mechanisms and strong sensor fusion and reinforcement strategies. The sensor information at the physical level is abstracted and organized into intermediate representations following a hierarchical organisation based upon increasing levels of persistence. As in (Sheshagiri & desJardins, 2002) these intermediate representations form the basis used to partition the deliberative process, defining regions of competences and dependencies among functional areas.

The `Reactive Controller` performs a simple closed loop behaviour arbitration strategy, querying the latest data processed from the sensors, feeding it to the `activities` and `behaviours` processing step and sending the controls suggested from the only active behaviour to the robot actuators when necessary.

Figure 4 provides an example of the principal components of the SRA Behavioural Suite which has evolved over the last 5 years of experience by the project's robotic research team. The suite reflects the result of an ongoing effort in identifying a set of navigation primitives and shows the relation between behaviour implementation, their characteristic parameters (in brackets) and the relevant events (showed in the rectangular diagram callouts) that may be notified to the Deliberative Layer during the execution of each behaviour.

The `Behaviour` base class offers standard parameters allowing the limitation of the maximum velocity of the robotic platform and the application of other constraints based on distance travelled and time elapsed from the initiation of the behaviour. More complex behaviours, such as `obstacle avoidance` and `wall following`, are implemented upon seminal methods for real time mobile robot obstacle avoidance like the Vector Field Histogram Plus [VFH+] (Ulrich and Borenstein, 1998) and the Dynamic Window algorithm (Fox et al, 1997).

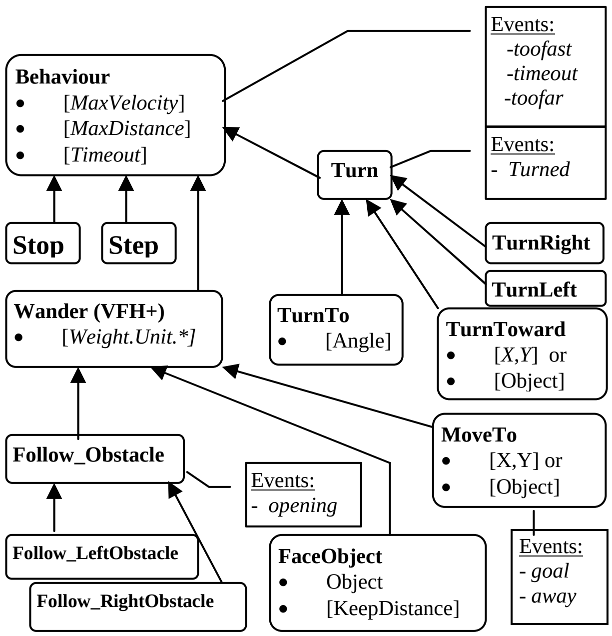

Figure 4: Social Robot Architecture Behavioural Suite

Some intermediate products of these behaviours are the *`obstacle histogram`* summarising the disposition of the obstacle in the vicinity of the robot and the *set of feasible directions*. The latter is found examining all the manoeuvres available in the robotic platform excluding those leading to a collision within a pre-determined timeframe. This information is used to define conditions (i.e. `BELIEF(narrow_ passage)`) fed to the reasoning process.

Another example of intermediate representation is given by the `Object Tracking` subsystem, which handles symbolic cursors similar to Saffiotti's work on perceptual anchoring (Coradeschi and Saffiotti, 2001). This subsystem emerges from the interaction between different object recognition components and the *`Object Tracking`* activity. The sensor information enters these object recognition components where different forms of feature extraction are employed on data coming from the sonar, the 3D laser range finder, and the vision system. These resulting feature hypotheses are then fed through a common API (`Observation(object_name,sensor_modality,…)`) to the `Object Tracking` activity in order to support the confidence building mechanisms employed for robust sensor fusion. This allows a general API framework to be employed for differing sensor modalities. The `Object Tracking` module deals with the reinforcement of the evidence values and the maintenance of symbolic cursors used by the behaviours (i.e. for way-point navigation) and by the deliberation process. In so doing, the `Object Tracking` sub-system also supervises the translation from quantitative data (i.e. the position of an object) to qualitative information (i.e. `BELIEF(close(object)),BELIEF(moving(object))`).

This strategy is the key mechanism for achieving a coherent integration of reactive and deliberative control strategies in autonomous mobile robots. This constitutes a fundamental component in bridging theoretical reasoning processes and the physical environment from the perspective of a stronger interpretation of embodiment as discussed in (Duffy, 2003).

Deliberative Level

The deliberative level of the Social Robotic Architecture follows a multi-agent-system (MAS) organization with several agents supervising the different functional levels of the robot. At any given time, a number of agents share the control of the robotic platform. These agents vary in complexity from simple procedural knowledge modules that deal with lower level capabilities of the platform (i.e. sensorial organization, configuration and behavioural arbitration), to means-ends reasoning.

The reasoning capability is delivered through Agent Factory (www.agentfactory.com) (Collier, 2001), an integrated and tooled environment for the rapid prototyping of social intentional agents based upon Belief-Desire-Intention (BDI) agent theory (Rao & Georgeff, 1995). BDI agents have been widely employed for the control of robotic systems as they facilitate the grounding of traditional symbolic reasoning in situations requiring real-time reactivity. In BDI systems, the deliberative layer is generally modelled using mental attitudes representing the information, motivational, and deliberative states of the agents. The role of these attributes is to provide the agent with a usable description of the present and future states of the agent's environment. Like other BDI implementations, Agent Factory agents employ practical reasoning techniques to deliberate upon their perceived situation, update their mental state and select the future line of action.

An important peculiarity of Agent Factory is the flexibility in which its cognitive agents can be attached to such diverse applicative domains through `Actuators` and `Perceptors` modules that integrate the abstract interface with different forms of embodiment. In the robotic domain we have used these characteristics to implement hardware and sensor abstractions. Agent Factory also facilitates the instantiation of robotic controllers and allows the installation of any combination of actuators and perceptors.

Agent specifications are stored in ASCII files containing Agent Factory Agent Programming Language (AF-APL (Ross et al, 2004)) scripts. AF-APL scripts contain initial beliefs; the declaration of actuators and perceptors in use by the robotic agent together with commitment rules governing behavioural transitions, plan activations, and goal decomposition.

The Platform Manager Agents constitutes the main script, which describes the robotic agent and supervises its initialisation. This script can also contain a list of references to additional AF-APL scripts (i.e. roles and plans), each specifying the BDI design for a different functional area. When the agent is initiated, these specifications are parsed, and the Java reflection API is used to instantiate an object for every actuator and perceptor class used.

Once the agent layer is initiated, a reserved belief (`startup`) is added to the belief set triggering a special actuator (`SetupActuator`) responsible for installing sensor drivers, activities and behaviours in the reactive controller. Figure 5 contains some of the design and the essential components of such a startup section for a robotic agent employed in a defensive role in the soccer scenario (see also the section detailing the SRA implementation).

```
// declare actuators and preceptor classes
ACTUATOR    SetupActuator,
ActivateBehaviorActuator, …
PERCEPTOR   RequestPerceptor, …

// ----  begin initial beliefs -----
// The home position of the left defender is 3
meters behind and 1.5 meters left of the center
pitch
// The home position of the attacker is 2
meters in front of the center pitch
Belief(Home(left_defender,-3,-1.5)),
Belief(Home(attacker,+2,0))
…
// Assume a specific role when requested
Belief(Requested(cover_role(?R)) =>
Commit(adoptBelief(role(?R))

// setup the reactive layer on startup
BELIEF(startup)
=>   Commit(setup_drivers(NomadScoutImpl,
PolaroidSonarRing, VirtualSonar , CCDCamera),
              setup_behaviours(MoveTo,
Intercept, DefendGoal))

// When requested to move to home position,
move  toward the home location for  the current
role
BELIEF(requested_achieve(reset) &
BELIEF(role(?R)) & BELIEF(Home (?R,?X,?Y)
=>  Commit(ActivateBehavior(MoveTo(x,?X,y,?Y)))
```

Figure 5. Part of the BDI design for a defender robot in the soccer scenario.

The example also illustrates the syntax required for the instantiation of a behaviour together with the specification of values for the parameters controlling its execution. For instance, based on the following commitment:

```
ActivateBehavior(BehaviorA((MaxDistance,2000,Tim
eout,1000,MaxVelocity,25))
```

the agent initiates a behavioral mode where the mapping between sensors and commands will be controlled by the specified behaviour class (named `BehaviourA` in the example). The same commitment also specifies the constrains for the behaviour like the maximum allowed velocity (25cm/sec), the maximum allocated time (1 second) and the maximum distance allowed to travel from its initiation as per the system's odometry feedback. The example activates all the constraints. Whenever one of these constraints is violated by the active behavior, the appropriate condition is added to the controller state and the correspondent beliefs (i.e. `BELIEF(toofast)`, `BELIEF(timeout)`, `BELIEF(toofar)`) is notified to the deliberative layer on every state transition.

Special parameters are reserved for the `Object Tracking` subsystem. This activity recognizes parameters defining conditions that may be monitored by the subsystem. These conditions describe the quantitative-qualitative mapping forming the base for basic spatial reasoning capabilities and controlling the interleaving between behavioral control and deliberation. For example, the following commitment:

```
COMMIT(Object_Tracking::set_condition(?Object.
distance<1000,close))
```

specify a mapping between the value of estimated distance to an object (i.e. the ball) and the subjective belief that the object is close to the robot. In the example the `Object tracking` subsystem will start monitoring the estimated distance raising the belief *`BELIEF(close)`* whenever such distance fall below the 1 meter.

This mechanism allows the agent to define interesting situation whose occurrence need to trigger the deliberation process based on subjective, task-dependent considerations. For instance, the robot in the example could have defined the threshold of one meter in order to regulate the transition between a `MoveTo` behaviour (i.e. aimed to move toward the ball while avoiding obstacles in the way) and a `FaceObject` behavior (i.e. aimed to establish a favorable position for a passage). In AF-APL syntax:

```
Belief(close(ball))) =>
        Commit( ActivateBehavior(
                MoveTo(Object, ball)))
Belief(close(ball))) =>
      Commit(…ActivateBehavior(
              FaceObject(Object,ball)))
```

The examples illustrate the mechanisms required to implement a BDI-based Deliberative Level with commitment rules controlling the behavioural level and determining the actions of the system. The following Social Level expands the framework presented so far to include explicit communication mechanisms for social interaction.

Social Level

Accordingly with the categories of agents outlined in (Molin & Chaib draa, 1996), Agent Factory agents are not only able to reason about themselves, but also about their acquaintances (those agents they encounter). These agents are thus able to select their goal, reason about how to satisfy that goal (by planning) and modify their own conduct based on circumstances and acquaintance behaviours.

To enable collaboration among social robots, Agent Factory developed agents make use of Speech Act Theory (Searle, 1969), a formalism for accurate and expressive communication mechanism in Multi-Agent Systems. This is undertaken by performing a speech act (such requesting, ordering, informing or promising) that sends a message to one or more of their socially capable acquaintances in order to affect their mental states. In this work, the robotic agents interact via the Agent Communication Language (ACL) *Teanga* described in (Rooney, 2000).

At a simple level, the received ACL messages may trigger specific commitment rules governing the reaction of the receiving agent. Extracted from the previous example in figure 5, the following illustrates how when the user is asked to move to its home position (`reset`) on the football pitch, it adopts the appropriate commitment.

```
BELIEF(requested_achieve(reset) &
BELIEF(role(?R)) & BELIEF(Home (?R,?X,?Y)
=>  Commit(ActivateBehavior(MoveTo(x,?X,y,?Y)))
```

Together with the language, the SRA implements a number of more sophisticated interaction protocols responding to the semantics described in FIPA specifications, among them, the Contract-Net-Protocol for group formation (http://www.fipa.org).

We argue that specific tools are needed in order to work toward true interoperability and cooperation between social heterogeneous robots (real, simulated & virtual) and effective human-robot interaction. The Virtual Robotic Workbench (VRW) is an instance of one such tool.

The Virtual Robotic Workbench

The Virtual Robotic Workbench provides a medium for experimentation in interoperability and cooperation between heterogeneous robots (real, simulated and virtual) and humans. As robots become more and more diffuse and employed in every-day tasks, some of the important issues will be the mediation between different forms of intelligence and autonomy and the assimilation of autonomous robotic entities within ubiquitous computing infrastructures. For these reasons the Virtual Robotic Workbench (Duffy, 1999; Mauro, 2005) was developed to ease configuration and networking for large scale heterogeneous real and virtual robotic teams, thus enabling dynamic composition of sensors and actuators and supporting dynamic discovery of resources and peers. In addition, the VRW offers a framework to support social integration, implemented in a re-usable and standardised form. The core features of the Virtual Robotic Workbench are the immersion of robots in a shared collaborative environment and the adoption of mature Multi Agent Systems technology in order to enable robot-robot and human-robot interaction.

The Virtual Robotic Workbench harvests seminal research conducted by the Collaborative Virtual Environment (CVE) and the Distributed Interactive System (DIS) communities. Each activity is seen as a collaborative experience in which information about activities (i.e. telemetry/ sensor data) and interaction between all participants (robots, experimenters and environment) are collected and shared, to varying degrees, within the framework, in a manner mimicking the diffusion of real events through the normal sensorial apparatus. In our solution, the connectivity to the shared environment is embedded in the sensing and actuating functionality of our robotic agents. By altering a robot's embodied format and establishing an interface to a standardized software agent layer within the architecture, the Virtual Robotic Workbench has facilitated the investigation of a new set of possibilities for interaction and interoperability in robotic contexts. These include indirect communication through augmented sensing, cooperation based on standard protocols (i.e. Contract Net for team formation) and migration or mutation of mobile agents among different robotic platforms.

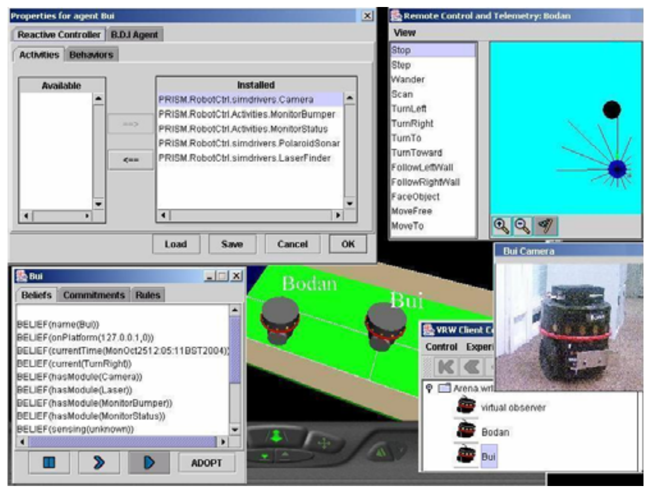

Figure 6. Workbench GUI tools, from upper-right clockwise: telemetry and remote control panel, video camera feedback, Workbench Console and Player tool, Agent design module, BDI debug panel and Agent properties panel. In Background: 3D viewer.

The Virtual Robotic Workbench offers:

- A communication medium - based upon a XML multicast protocol - which is exploited for dynamic resource discovery and to exchange information and control among humans and robots. This messaging infrastructure supports synchronized streams of sensorial data originating from the robots, together with agent-like directives encapsulating robot-robot and robot-human interactions.
- A visualisation medium (Virtual Robotic Workbench Visualisation Suite) which offers real-time, multimedia visualisation facilitating behavioural scrutiny and situational awareness for humans involved in, for example, such tasks as semi-autonomous remote control,

supervision and inspection of large teams of mobile robots.

- A FIPA (http://www.fipa.org) compliant Agent technology which supervises the social interface between each user and the shared environment. This technology is employed both for the control of the robotic platforms and, in the context of personal assistants, embedded in the human user interface. In each case, this agent component encapsulates social aspects of the teamwork like acquaintances maintenance, team formation and coordination. From the user point of view, this component acts as a proxy for the human, handling data presentation and helping in conducting user policies during team operations.

The Virtual Robotic Workbench Visualization Suite can deliver a dynamic, interactive and multimodal visualization of activities in the shared environment. Furthermore, the Workbench can also transmit a dynamic 3D scene across the network where robots are represented as robotic avatars and their associated movements as transformations within the 3D world.

In addition, to superimpose virtual objects adequately on a real world scene to achieve augmented reality, the precise on screen coordinates have to be known. Therefore, in conjunction with the Virtual Robotic Workbench, we use ARToolkit, a widely utilised library for tracking markers in a video image, in order to gather the location of a robot and for example superimpose an avatar on top of it. The mixture of reality and virtuality allows researchers to conduct experiments with only a few real robots, which supply the system with real-world data – as opposed to a pure simulation – and numerous virtual robots, that would be otherwise infeasible due to cost or maintenance. A CyberMind Head Mounted Display is used for visualisation.

The following section demonstrates how these various technologies are integrated.

Implementation

We currently avail of a range of different robots, including *Nomad Scouts* (two-wheel differential drive robot manufactured by Nomad Technologies), *ER1* from Evolution Robotics, and two custom physical humanoids (“Anthropos” and “Joe”) and virtual robot avatars. The robots are equipped with various (and changeable) combinations of sensors like ultrasonic range scanners, bumpers for collision detection, CCD cameras, odometers and laser range finders. For this purpose the Social Robotic Architecture comprises a suite of device drivers, behaviours and task definitions based upon an abstract physical interface.

The following sections illustrate how the Social Robot Architecture has been instantiated on a range of different platforms which demonstrate its range of universal applicability.

The Real, the Virtual and the Human

In the example, on start-up, the drivers for the Nomad Scout II platform, the Polaroid sonar ring and the CCD Camera are all installed as activities in the reactive layer together with a virtual module (VirtualSonar) responsible for sensing simulated robots. At every cycle of the controller, each activity is requested to process its input independently from other modules. The output of each activity is cascaded to the next activity based on a pre-determined order defined on start-up - it is impossible to define dependencies other than this simple chaining mechanism. The final output is then analyzed by the active behaviour. This scheme, simple though it is, nonetheless makes it very easy to combine standard modules, operating with the physical robotic board and real sensor drivers, with virtual modules.

This proves a key feature of the architecture which allows fast and accurate prototyping of new versions of robotic agents. We can predict more accurately what would happen if a perceptor or any component of the reactive controller were modified. Usually it is easier to change virtual modules then normal sensors (which involve interfacing with physical drivers) so that it is a real advantage testing the robotic controller and replacing the component under development with his virtual counterpart. Through the ability to mix real and virtual components, we can test, for example, what would happen if we changed the characteristics of the sonar ring (e.g. the location of the emitters) or alternately adopt a new type of range sensor (e.g. lasers) running a real robotic agent with all the standard components, but rather with the simulated sensor.

Robot Soccer Anywhere

We have developed a behaviour library for *Robot Soccer Anywhere* (Dragone et al, 2004), an open framework for people to play football with a number of robots in free environments. Skills like kicking, dribbling, passing, and following the ball are implemented. Individual and team strategies, goal-oriented behaviours and planning are delivered through the intentional agent structures of Agent Factory (Collier, 2001), using modal logical rules and `Plans` in the Belief Desire Intention Deliberative Layer.

Figure 7: An augmented reality virtual head used for human-robot social interaction while playing Robotic Soccer Anywhere

Real and Virtual Robot Waltz

Another experiment developed to demonstrate the fusing of reality and virtuality is the mixed waltz, wherein a real and a virtual robot collectively perform a simple manoeuvre (the box waltz). Both robots coordinate their movements by sending messages to each other through the Social Level of the Social Robot Architecture while using the Reactive Level to avoid obstacles, be they physical or virtual. Figure 8 provides a snapshot of a real robot successfully waltzing with its virtual counterpart.

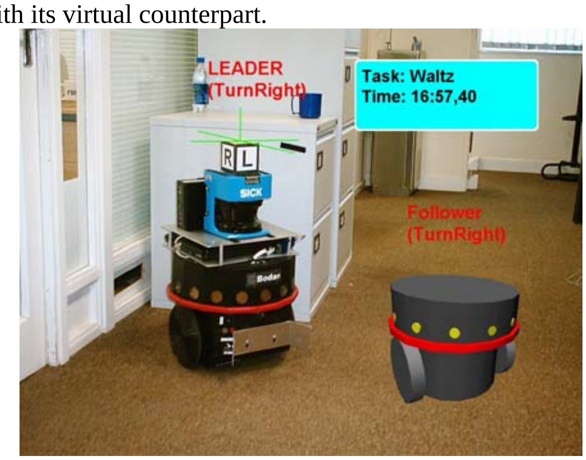

Figure 8. Augmented Reality Visualization of the Mixed Waltz task

Humanoid-Robot interaction

The Social Robot Architecture was also implemented on the humanoid robot “Joe” in order to demonstrate directed voice-based human-robot interaction. The ViaVoice (IBM) speech system was used for conversation based communication in conjunction with the Agent Communication Language: Teanga (as similarly implemented in work by Jacobus and Duffy (2003)).

Figure 7: The social robot “Joe”

The utterance types, based on Teanga's speech acts, consist of `query`, `inform`, `commit`, and `declare`, and their subtypes, with shorter versions of commonly used subtypes being optional. These combined to deliver the content, which consisted of an action type, a property, or an action modified by additional `properties`. These `properties` describe the state of the variable or object they are mapped to, and can be required parts of action types or expressed independently as an assertion of belief. Examples of implemented speech acts include:

```
REQUEST     target:Joe
            action:list_files
            property:aloud
REPLY       target:human-user
            property:[filelist]
REQUEST     target:Joe
            action:open_file
            property:[filename]
REPLY       target:human-user
            property:[filename]
```

Given that the social interaction between the human and the robot is goal-directed, the success of the system is viewed as the completion of a stand alone task or as a step in a more global task objective. Preliminary usage of a limited vocabulary has shown a highly accurate recognition and action performance by the robot, even when commands are initially given with incomplete information such as failure to provide filenames. Human familiarity with the ACL syntax is the factor with the greatest effect on the percentage of successful interactions, which emphasises the need for the ACL to support the clarifications of partial or malformed commands.

Conclusions

The core objectives of the Social Robot Architecture have been to develop a robust multi-layered BDI architecture for explicit social interaction between a robot and either one or more other robots, virtual avatars or humans. It has been designed to cope with many different hardware constructions through its abstraction of specific low-level hardware issues.

The Social Robot Architecture in conjunction with the social analogies of *identity*, *character*, *stereotypes*, and *roles* (not discussed in this paper but detailed in (Duffy, 2004)), has resulted in a control methodology that develops, implements and tests strong social robot concepts. In so doing, this work has aimed at addressing inadequacies of existing multi-entity research and applies the developed concepts to both the real world and augmented reality domains. The Social Robot Architecture constitutes the successful implementation of a cohesive reactive-deliberative synthesis of BDI-based deliberation methodologies in a real world social context.

We have briefly illustrated the SRA in three different scenarios. These have been developed to show behavioural hierarchy, the fusion between reality and virtuality, and the architecture’s communication capabilities respectively. Online video demonstrations of this work can be found at http://www.cs.ucd.ie/csprism.

Acknowledgments

This research is based upon works supported by the Science Foundation Ireland under Grant No. 03/IN.3/1361.

References

Arbib, M. (1992), Schema Theory, in S. Shapiro, ed., *The Encyclopedia of Artificial Intelligence, 2nd Edition*, Wiley-Interscience, pp. 1427-1443.

Arkin, R. C. (1987), Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming for Behavior, in *IEEE International Conference on Robotics and Automation*, Raleigh, NC, pp. 264-271.

Arkin, R.C., Carter, W., MacKenzie, D.C. (1993), “Active avoidance: escape and dodging behaviors for reactive control”, *Active Robot Vision*, World Scientific Publishing, pp.175-192.

Billard A., Dautenhahn, K. (1997) “Grounding communication in situated, social robots”, *Proc. TIMR97 "Towards Intelligent Mobile Robots"*, published in the Technical Report Series of the Department of Computer Science, Manchester University

Brooks, R.A. (1986), 'A Robust Layered Control System for a Mobile Robot', *IEEE Journal of Robotics and Automation* RA-2, 14-23.

Brooks, R.A. (1991), Intelligence Without Reason, in 'Proceedings, IJCAI-91', Morgan Kaufmann Publisher, Inc., Sydney, Australia, pp. 569-595.

Brooks, R.A., (1991) “Integrated Systems Based on Behaviors”, *SIGART Bulletin, Vol. 2*, No. 4, August, pp. 46—5

Cai, A, Fukuda, T., et al. (1995) “Hierarchical Control Architecture for Cellular Robotic System – Simulation and Experiments”, *IEEE International Conference on Robotics and Automation*

Collier, R.W., Agent Factory: A Framework for the Engineering of Agent-Oriented Applications, *Ph.D. Thesis*, University College Dublin, Ireland, 2001

Connell, J. H. (1992), “SSS: A hybrid architecture applied to robot navigation," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2719-2724.

Coradeschi S., Saffiotti, A. (2001), Perceptual Anchoring of Symbols for Action. *Proc. of the 17th IJCAI Conf*, 407-412. Seattle, WA.

Dautenhahn, K. (1995), Getting to know each other - artificial social intelligence for autonomous robots, Robotics and Autonomous Systems 16, pp 333-356

Donald, B., Jennings, J., Rus, D. (1997) “Minimalism + Distribution = Supermodularity”, *Journal of Experimental and Theoretical AI, 9* (20-3), pages 293-321

Dragone, M. O'Donoghue, R. Leonard, J.J. O'Hare, G.M.P Duffy, B.R. Patrikalakis, A. Leederkerken, J. (2005) “Robot Soccer Anywhere: Achieving Persistent Autonomous Navigation, Mapping and Object Vision Tracking in Dynamic Environments”,. In Proceedings of SPIE Opto Ireland, Dublin, Ireland, April

Dragone, M. (2005) “The Virtual Robotic Workbench”, *Technical Report No. UCD-PRISM-05-11*, Department of Computer Science, University College Dublin

Drogoul, A., Ferber, J., (1992) “Multi-agent simulation as a tool for modelling societies: Application to social differentiation in ant colonies”, *Workshop MAAMAW'92*.

Duffy, B.R., Garcia, C., Rooney, C.F.B., O'Hare, G.M.P., “Sensor Fusion for Social Robotics”, *31st International Symposium on Robotics (ISR2000)*, May 14-17, 2000, Palais des congrès, Montréal, Canada

Duffy, B.R., (2004), "Social Embodiment in Autonomous Mobile Robotics", *International Journal of Advanced Robotic Systems*, 1 (3), pp.155-170.

Duffy, B.R., (2003) "Anthropomorphism and The Social Robot", *Special Issue on Socially Interactive Robots, Robotics and Autonomous Systems 42* (3-4), 31 March, pp170-190

Duffy, B.R., O'Hare, G.M.P., O'Donoghue, R.P.S., Rooney, C.F.B., and Collier, R.W (1999) "Reality and virtual reality in mobile robotics", *1st International Workshop on Managing Interactions in Smart Environments MANSE'99*, Dublin December

Fox, D., Burgard, W., Thrun, S., “The dynamic window approach to collision avoidance”, IEEE Robotics and Automation Magazine, 4(1), March

Goldberg, D., Cicirello, V., Dias, M.B., Simmons, R., Smith, S., and Stentz, A. (2003) "Market-Based Multi-Robot Planning in a Distributed Layered Architecture," *Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the 2003 International Workshop on Multi-Robot Systems*, Kluwer Academic Publishers, Vol. 2, pp. 27-38

Grey, W., The Living Brain, (1953), Penguin, London, 1967

Jacobus, E., Duffy, B.R., (2003) "The Language of Machines", *Digital Interaction, International Symposium on Information and Communication Technologies*, 24-26 September, Trinity College Dublin, Ireland

Kortenkamp, D. Bonasso, R.P. and Murphy, R. eds., (1998), *Artificial Intelligence and Mobile Robots*, AAAI Press

Kuniyoshi, Y. (1997), “Fusing autonomy and sociability in robots”, *International Conference on Autonomous Agents*, pp.470-471.

Logan, B. Sloman, A., (1998) “Cognition and affect: Architectures and tools”, *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, ACM Press, 1998, pp 471—472

Mataric, M.J. (1990) Environment learning using a distributed representation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 402-406

Mataric, Maja J, (1997) “Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior” *Journal of Experimental and Theoretical Artificial Intelligence*, special issue on Software Architectures for Physical Agents, 9(2-3), H. Hexmoor, I. Horswill, and D. Kortenkamp, eds., 323-336

Molin B., & Chaib Draa, B., (1996) “An overview of distributed artificial intelligence, Foundations of Distributed Artificial Intelligence, O’Hare & Jennings (eds).

Nakauchi Y., Simmons. R. (2002) “A Social Robot that Stands in Line”, *Autonomous Robots, 12*:3 pp.313-324, May

Oka T., Inaba M., Inoue, H. (1997), “Describing a modular motion system based on a real time process network model”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.821-827.

Parker, L.E., (1998) “ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation”, *IEEE Transactions on Robotics and Automation*, *14* (2),

Rao, A. S. and Georgeff, M. P. (1995), “BDI Agents: From Theory to Practice”. *1st International Conference on Multi-Agent Systems (ICMAS'95)*, San Francisco, CA, 312–319.

Rooney, C., (2000) “A formal semantics for Teanga”, *Technical Report No. UCD-PRISM-00-05*, Department of Computer Science, University College Dublin

Ross, R, Collier, R, O Hare, G.M.P. (2004) “AF-APL: Bridging princples & practices in agent oriented languages”, *In Proc. The Second International Workshop on Programming Multiagent Systems Languages and tools (PROMAS 2004)*. Held at AAMAS 04, New York, USA,

Searle, J. (1969) Speech Acts: An essay in the philosophy of language, London, Cambridge University Press.

Sequeira, J., Ribeiro, M.I., (1998) “Rigid Bar Transportation by Two Mobile Manipulators: A Behaviour-Based Cooperative Approach”, *Proceedings of the 3rd.Portuguese Conference on Automatic Control, CONTROLO'98,* Coimbra, Portugal, September, pp.661-666

Sheshagiri M., & desJardins, M., (2002) Data Persistence: A Design Principle for Hybrid Robot Control Architectures, Paper, *International Conference On Knowledge Based Computer Systems*, Mumbai, India 2002

Tambe, M. (1997), “Agent Architectures for Flexible, Practical Teamwork”, *National Conference on Artificial Intelligence* (AAAI-97)

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Creemers, A.B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D. (2000), “Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva”, *International Journal of Robotics Research, Vol. 19*, No. 11, November, pp. 972-999.

Ulrich I., Borenstein, J. (1998), “VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation.* Leuven, Belgium, May 16–21, pp. 1572 – 1577