Conversational Sales Assistants

Frank Guerin, Kaveh Kamyab, Yasmine Arafa, and Prof. Ebrahim Mamdani

Intelligent and Interactive Systems
Department of Electrical & Electronic Engineering,
Imperial College of Science, Technology & Medicine,
Exhibition Road, London, SW7 2BZ.
+44 20 7594 6331

f.guerin@ic.ac.uk, k.kamyab@ic.ac.uk, y.arafa@ic.ac.uk, e.mamdani@ic.ac.uk

ABSTRACT

Designing believable embodied agents requires an awareness of context and the capability to communicate with speech and gestures. We are concerned with developing such agents for use as sales assistants in a virtual marketplace. We employ a communication model which takes a public perspective on the meaning of communicative acts to enhance the agent's social awareness. Additionally the agent draws private inferences to build a model of the user. The agent uses a BDI model to reason about its contextual information and plan communicative acts. The speech and gestures generated by the agent are integrated using a script language. We contend that the integration of these components will help us to achieve contextual embodied agents.

1. INTRODUCTION

We are concerned with developing embodied agents for use as sales assistants in a virtual marketplace. These agents must be capable of interacting in a realistic way and this requires an awareness of context and the capability to communicate with speech and gestures to enhance believability.

This work is being undertaken in the SoNG project (Portal**S** **o**f **N**ext **G**eneration). Current portals (For example: Yahoo, Excite and Netscape) present the user with a 2-D interface. It is envisaged that the next generation of portals will offer 3-D virtual environments, for example including virtual marketplaces and chat rooms. Users will be able to navigate their avatar through this virtual space and enter shops to inspect goods and interact with virtual or real humans to make transactions. The technology required to implement these virtual humans includes the realistic animation of synthetic faces and bodies as well as adequate conversational skills.

Our project partners have been working on designing the virtual world (based on the central square in Turin), the user interface, the facial and body graphics and animation (for synthetic characters) and the MPEG-4 interfaces. These synthetic characters will be driven by agent technology; the design of these agents is the subject of this paper. This is a work in progress, at this point most of the architecture has already been planned, but much of the implementation is still in progress. In this article we comment on our proposed architecture for the agents, in particular how the inputs to an agent are interpreted according to conversation protocols and how a user profile is built to model the user's preferences.

Our agents receive verbal communication as input, and also respond to the user's movements such as moving around the shop. The agent's responses include verbal communication enhanced with accompanying facial animations as well as communicative gestures (body movements such as pointing).

A short description of the content follows. In Section 2 we give a brief overview of the requirements for our sales agents to be effective in the virtual marketplace being developed for the SoNG project. In Section 3 we present our solution to the problem of interpreting communicative acts in context. Section 4 describes how the output communicative acts can be chosen when a conversation is following a predefined protocol. Section 5 describes a scripting language for integrates descriptions of body movement, speech output and emotional parameters. Section 6 describes the agent architecture used and Section 7 describes our implementation of these agents using the JAVA Agent DEvelopment Framework (JADE) and the Java Expert System Shell (JESS). Section 8 gives a brief summary and outlines issues that must be tackled in the future.

2. REQUIREMENTS FOR SALES AGENTS

We are concerned with using embodied agents as sales assistants in a virtual marketplace. Within the virtual marketplace of SoNG several demonstration applications shall be developed including a theatre booking system, a phone shop and a clothes shop. Here we focus on the telephone online shop. The sales agent must be able to converse with the customer, to respond to queries for phones satisfying certain criteria and to display phones to the customer. This application requires not only that the agents must have the relevant product information, but also they must be able to engage the customer in an interesting conversation. In order to sustain the user's interest the agent should be:

1. Believable. The agent must appear believable to the customer, using human-like gestures as well as verbal communication skills. At a lower level it will be necessary to synchronise gestures and verbal communication.
2. Proactive. The agent must proactively introduce new topics of conversation, for example describing new products. By the appropriate use of gestures the agent will appear exciting and enthusiastic about products.

3. Context Sensitive. During an interaction, the agent will draw inferences about the customer's preferences and build up a user model for each customer. Using this, it can tailor its conversation to be appropriate to the customer's interests. Additionally the agent must be aware of the social context of the interaction, e.g. an agent acting as a shopkeeper should be aware that it is obliged to greet a new customer and respond to queries.

There are two main body animations required for sales assistants: pointing at a product display and walking to a part of the shop. Implementation of these body movements requires integration with the 3-D objects of the virtual world (e.g. the agent needs to know where to point to). Facial animations are more numerous and include expressions of happiness or sadness, raising of eyebrows and a gesture denoting inability.

In addition the agent must be able to deal with natural language queries involving fuzzy terms such as cheap or lightweight. Since these terms are subjective the agent should build a user model of the customer in order to know what the fuzzy terms map to for each customer.

3. INTERPRETING COMMUNICATIVE ACTS

Through an online interface users may type text in natural language and may select gestures for their avatar to execute in the virtual world. The sales agent must interpret these verbal and non-verbal communicative acts so that it may update its model of the world and respond appropriately. The interpretation of an act does not depend on the act alone, communication can be highly context sensitive, i.e. there are certain external factors which affect the meaning of a speech act. These factors include the domain in which the conversation takes place, the status or authority of participants and the relationship an act has to the remainder of the discourse. We handle context in two ways:

1. Firstly we have an explicit representation of the context in the form of a *conversation state*, which holds information known to all participants in the conversation. The conversation state is a set of propositions (e.g. representing expressed mental attitudes[†]) and variables important to the conversation (e.g. roles occupied by participants). In our communication model we define the meaning of communicative acts as a function from context onto context [2] i.e. the conversation state contributes to the meaning of acts.
2. Secondly we make use of protocols to specify roles of participants and to encode information appropriate to the current domain. There are *social roles* which are permanently active and also roles within conversation protocols which last only for the duration of the conversation. The roles agents play in a protocol (or in the society) determine the social obligations of the agent in the conversation and also affect the meanings of acts.

We define semantics for events in the virtual world firstly from a public perspective (the meaning according to the conventions of the society) and secondly from a private perspective (the inferences that agent itself draws). The conventional (public) meanings of acts are predefined and common to all agents in the world. The public meanings are used to build the model of the conversation state. Thus there is a standard meaning for each communicative act and all participants can update their copy of the social state in the same way.

Private inferences are agent specific and will update the agent's own model of the world, in particular they will update the agent's model of the user it is interacting with. A user modelling component is an essential part of an e-commerce system such as the SoNG application. Its use is threefold: firstly to support the dialogue manager by modelling users' dialogue goals, secondly to model users' product preferences and finally to model users' interaction preferences.

In addition, modelling a user's product preferences allows the agent to tailor suggestions to the user's specific requirements and modelling a user's interaction preferences allows the agent to vary the intensity and quality of conversation and animation. Work by Charlton *et al.* [1] shows that a successfully employing these two models can greatly increase a user's perception of intelligence and trust in a system.

We now look at a simple example of the interpretation of communicative acts to illustrate how the public meanings and protocols update the conversation state. In our phone shop scenario, the sales agent permanently occupies the social role of "shopkeeper". A user's avatar entering the shop is an event which must be interpreted. The public meaning of this event defines that a new conversation state must be created and asserts two role variables and one proposition within that new state. The variables are: the agent that occupies the social role "shopkeeper" is now assigned the conversational role "shopkeeper" and the user's avatar is assigned the role "customer". The new proposition asserts the existence of a social obligation for the shopkeeper to greet the customer.

Subsequently the customer queries about the availability of WAP enabled phones. The public meaning of this query adds two new propositions to the conversation state. The first is that the customer has expressed a desire to know about the availability of WAP enabled phones and the second asserts the existence of a social obligation for the shopkeeper to tell the customer about WAP enabled phones. In addition to these public meanings the sales agent makes a private inference and adds to its user model for this customer a proposition describing the users preference for WAP enabled phones. This record of the user's preference will be useful in future interactions. For example if the customer returns at a later date, the sales agent will automatically check if there are any new WAP enabled phones in stock since the previous interaction and inform the customer.

[†] We distinguish between an agent's publicly expressed mental attitudes and its personal internal mental attitudes. These need not be identical, agents may not be sincere. An agent may express a desire to have an action performed if it does not really desire the action to be performed, it may be testing the willingness of another agent to comply for example. This means that an agent does not need to hold a mental attitude as a precondition to expressing it.

4. PLANNING COMMUNICATIVE ACTS

The sales agents need to plan their communicative acts in order to satisfy the requirements listed in section 2. Protocols are defined for each interaction, and these constrain the agent's choices at any point in the conversation. The choices made are determined by the propositions existing in the agent's mental state. In particular the agent's model of the user plays a large part. For example, Figure 1 shows a UML-style statechart diagram describing a protocol for an interaction between a sales agent (playing the role of Shopkeeper) and a human user (Customer). Agents keep a record of the conversation's state and use this to plan communicative acts, hence we use a statechart diagram. In state 3 the Shopkeeper may decide to give information on a new product based on knowledge (from the user model) of the product the user is interested in.

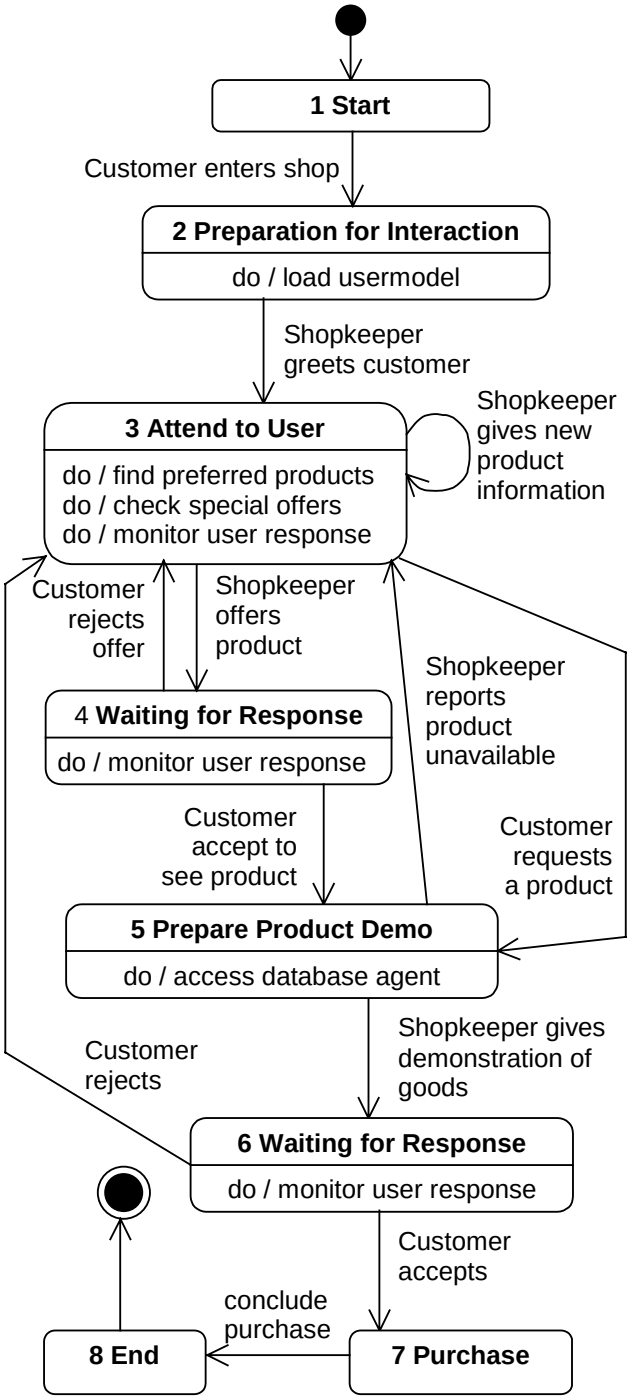

Figure 1: Shopkeeper-Customer Protocol

Once the sales agent has decided which path to take in the protocol, the agent has a certain characteristic behaviour, and this determines the gestures that will be chosen to accompany the speech output. The action planner attaches an absolute starting time to each action (speech or gesture) this enables the rendering system to synchronise the different modes of expression.

5. LINKING MULTIMODAL OUTPUTS

When the agent needs to control its synthetic character in the virtual world it generates an instruction describing the necessary action. From this instruction a CML (Character Mark-up Language) script is generated. CML is an XML-based scripting language which integrates descriptions of body movement, speech output and emotional parameters. Character actions are grouped into five categories:

1. Movement, e.g. dancing, walking and running.
2. Pointing, e.g. with open hand, index finger, or while holding the object.
3. Gestures, e.g. flicking of the hand or a polite sweep of the arm for welcoming.
4. Handling objects, e.g. to grab an object or raise it.
5. Speech, including pitch and intonation parameters.

Tags are available for each of these actions. The parameters describing the action are placed between the tags, these may include the position to move to, the speed of movement and the angle to move through. The language contains low-level tags defining specific character gesture representations defining movements, intensities and explicit expressions. There are also high-level tags that can define commonly used combinations of these low-level tags.

For actions such as pointing it is necessary to refer to another object in the virtual world, in these cases a reference is made to the *assets* [6] description of the world. The *assets* description extracts all information about useful objects in the virtual world, for example the dimensions and position of avatars and objects. It also records the functional properties of objects so that the embodied agents know how to interact with them, for example it records if an object can be handled. An *assets* description is encoded with XML tags.

CML is designed to be sufficiently general to be used in any system, therefore it must be passed to a rendering system which translates the CML script instructions into the appropriate functions for the system being used. CML acts as a universal interface between embodied agents and rendering systems as shown in **Figure 2**.

The appropriate translations are described with XML Schema structures. These structure definitions are stored in a Schema Document Type Definition (DTD) file using XSDL (XML Schema Definition Language). At run-time character behaviour is generated by specifying XML tag/text streams which are then

interpreted by the rendering system based on the rules defined in the definition file.

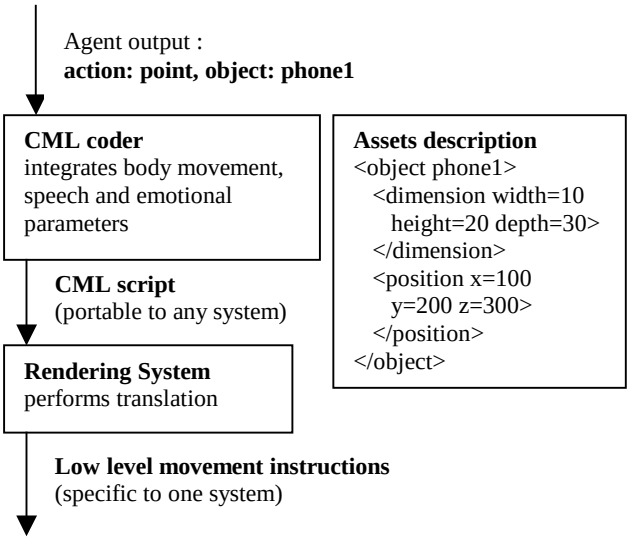

Figure 2 : Using CML

Now we take a look at a simple example: supposing our agent wishes to point to an object and say "here is our new WAP phone":

1. The agent generates the instructions:
   action: point, object: phone1, behaviour: enthusiastic, time: 01:52:18, duration: 1
   action: speak, text: "here is our new WAP phone", behaviour: enthusiastic, time: 01:52:16

2. The CML coder produces the script which describes both these actions and integrates the emotional behaviour with the spoken text and the pointing action. For the text, certain tags are inserted in the string which describe the pitch and tone e.g. "^p(0.9)". For the pointing action, the speed, duration and angle of movement are chosen to express the desired emotional parameter (enthusiasm).

3. The rendering system translates the CML script into instructions specific for the system in use. For example, the pointing action may translate to a description of the movement of individual joints of the body.

Different rendering systems may implement the CML script in different ways, for example some parameters such as duration may not be available on some systems and they will have to be ignored.

6. SYSTEM ARCHITECTURE

SoNG agents are situated in a virtual world and interact with the world through the MPEG-4 player. For example, to find about changes in the world, (such as a user speaking or moving) and to effect changes (facial and body animations for the agents virtual body). We have developed a general agent architecture to be used for generating agents that communicate multimodally.

There are three different agents involved in a typical interaction with the user. The user sees only one, the sales agent, which is embodied. The user may ask for a product in natural language, this request is forwarded to a search agent who queries one or more database agents and returns with the required information. The architecture of the agents themselves is modular thus allowing each of our agents to be built with re-usable components. For example, both embodied (sales agents) and dis-embodied (database) agents will use the same module for interpreting the meaning of events in the world, but embodied agents will use a natural language processor while disembodied ones will not.

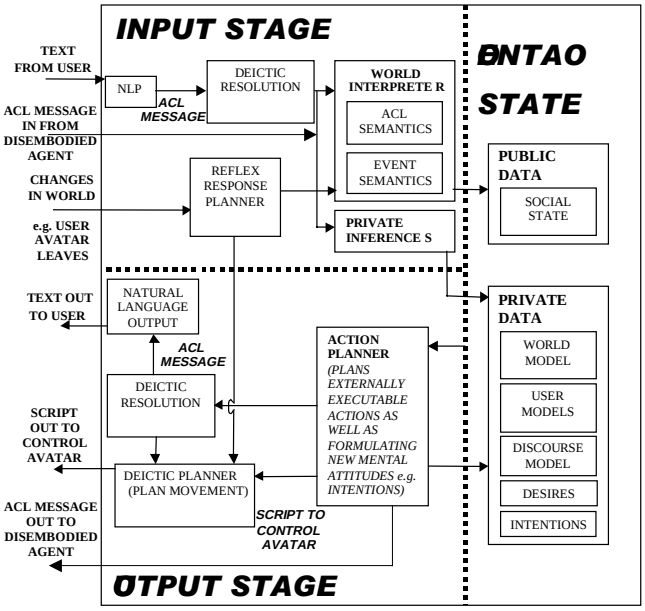

Figure 3: Agent Architecture for Embodied Agents

As depicted in Figure 3, the agent architecture is divided (by heavy dotted line) into input (top left) and output (bottom left) modules and an internal mental state (right). The mental state is updated by the input stage's interpretation of events in the virtual world. The output stage uses the current mental state to decide what actions it should do next. In addition to selecting actions, the action planner may also modify the mental state as it may decide to add a new desire for example.

An aim of the design is to raise the level of agent-agent interactions. Note that any communication from a human to an agent is passed through the natural language processor and converted to an ACL message. From this point on, it is processed in the same way as a message coming from a synthetic agent. In this way the agent-agent communication is at the same level as human agent communication, i.e. it is a high level communication, not just the exchange of procedural directives.

6.1 Input Stage

On the input side, agents situated in a virtual environment observe relevant changes in the virtual world. These include events such as communicative acts (speech acts and gestures) and movements of other agents in the world. We distinguish between private inferences and public inferences, thus events are interpreted according to their conventional meaning as defined by the social conventions (implemented in the world interpreter) and additionally according to the private inferences an agent makes.

6.1.1 Deictic Resolution

The Deictic resolution [5] handles references to objects in the world, for example if a user says "that phone" and points to a phone, it will resolve the reference. Similarly, in the output the agent may refer to a particular phone model, but the deictic resolution may replace this with "that phone" and instruct the deictic planner to perform a pointing gesture. The reflex response planner deals with movements in the world, for example, the sales agent's gaze should follow the user as he moves around the shop.

6.1.2 World Interpreter

The world interpreter gives the agent its social awareness. It is here that the semantics of communicative acts, events and protocols are stored. These semantics are viewed from a public perspective [9] so that they describe the conventional meaning in the society. The semantics are encoded as rules which add new propositions to the agents model of the social state. For example, if a user asks for a product, a proposition asserting that the user has expressed a desire for a product will be added to the social state.

6.2 User Modelling and Personalisation

Currently SoNG agents support an explicit model of users' domain specific goals and knowledge and product preference. These are held in the form of beliefs stored by the user model. User interaction preferences will be modelled at a later stage.

Although users' goals and knowledge are modelled by "crisp" facts in the agent's knowledge base, product preferences are associated with a preference strength modelled by a fuzzy set. Strengths can vary in a range of 0 to 1 and are associated with linguistic terms such as "like" and "dislike".

The rule-based inference mechanism is capable of making inferences about a newly expressed user preference or updating preferences already existing in the user model. The user's strength of preference for a product can be updated so that the current value is an average spanning many interaction sessions. A set of rules is used to update the values of fuzzy variables representing user preferences.

When an agent learns about a user's preferences it is then able to adapt its dialogue plans to suit these preferences. For example, if a user asks for a generic phone AND the agent believes the user likes Nokia phones, then the agent will suggest a Nokia.

6.3 Mental State

The agent's mental state makes use of the BDI [8] architecture; i.e. the agent has a set of beliefs, desires and intentions. Among the agent's beliefs are beliefs about the beliefs, desires and intentions of others (for example in the user model). The social state contains data arising from the conventional meaning of events as described in the world model. This data includes expressed mental attitudes of agents and obligations to perform certain acts. The agent's know-how is also included in its initial beliefs as a set of rules where the left hand side is a desired goal or sub goal and the right hand side is an intention or a set of further sub goals. The typical operations performed on the mental model during an agent cycle involve translating desires for into intentions for actions by using the knowledge encoded in the beliefs.

6.4 OUTPUT STAGE: ACTION PLANNER

The Action Planner is the module responsible for interpreting inputs from the Social Model and formulating an output script. Inputs can be interpreted as interactional or propositional conversational functions. Interactional functions can include such inputs as "Excuse me" or "No, no, I meant…". All other inputs will be of the propositional type. A user's conversational intentions include queries (where, when, how, how much, which, can I, etc.), references to objects (that, this, those, these, etc.) as well as simple conversational interaction.

The planning module is implemented by a set of planning rules. It does not formulate a complete plan in advance as a sequence of actions for each step it will take, rather, it just applies its rules at each stage and selects the next action. The right hand side of these rules is the adoption of a new mental attitude by the agent. It may be an intention to perform an external action (for example, send a speech act or a script) or it may be the assertion of a proposition which does not map directly to any external action, but will cause other rules to fire in subsequent iterations of the planning process. We are developing an internal language with which agents reason. For example an agent may assert the belief that he is attempting or has done an action, or an agent may intend to *findif* or *tellif* with respect to some proposition. Italics here denote the predicates of the internal language, these appear in the left-hand side of rules that will fire subsequently.

The Action Planner will utilise a model of the current conversation or discourse model. Each agent will keep one discourse model per user containing objects referenced, queries made and products bought. The Action Planner will be able to use the model to maintain a context for a conversation and further features such as referring to a past transaction. For example, if a user returns to a sales agent after having purchased a CD, the agent can ask, "How was the CD you bought last time?". The user's response will allow the agent to update its beliefs about the user's preferences. The Planner will manage turn taking during a conversation and will also be able to proactively initiate a conversation with a user.

Queries about domain information, such as the price of objects, etc., are forwarded to a search agent capable of retrieving up to date information from a database. Results from queries will be filtered and embedded into the conversation.

Output from the Action Planner will be in the form of a script describing high-level conversation functions. Each element of the script will be time-stamped for synchronisation. A CML script is then generated which ensures synchronization between verbal and nonverbal signals. The CML script is rendered to produce Facial Animation Parameters (FAPs) and Body Animation Parameters (BAPs) which are used to control the positions of the characters joints at a low level.

7. IMPLEMENTATION

The implementation of our agents is composed of a set of modular building blocks. At the lowest level they are built using JADE [3], an agent platform that provides, amongst other features, the infrastructure for agent communication. In SoNG we have extended the JADE agent class to define a class hierarchy of agents needed for our application. To implement internal planning and reasoning we use the JESS [4] to build a rule based action planner for each agent. JESS operates on a Knowledge Base

(KB), which represents each agent's public data (social model) and private data (beliefs about the state of the world, the user and the discourse, as well as desires and intentions). Each agent is able to make inferences about events in the environment and update its beliefs. Furthermore, each agent makes decisions according to the state of it beliefs. JESS uses a very simple mechanism to implement this functionality. A rule base is constructed and loaded into the JESS engine at runtime and rules are activated if a set of preconditions (or antecedents) is true. For example, if an agent believes that it must reply to a user query, and it has the necessary information to do so, then it will reply to the user with that information. In addition, JESS knowledge bases can be stored as data in relational databases such as those used to maintain product information in the e-commerce application.

We extend JADE's "agent" class with our "genericAgent" and use a cyclic agent behaviour. Our JADE agents each have their own *jess_processor* object. This is constructed during the agent's *setup* phase, it starts a new JESS *FuzzyRete* engine and loads all the rules appropriate for that agent. Each agent has rules stored as text files for inferences, planning, beliefs and desires. The agent's type variable prefixes the filename in each case, for example "phoneinferences", so that the appropriate files can be loaded. In addition there are common rules for each agent, these are the social model, mental model and semantics of communication and events. The Database agent is a special case, and its *jess_processor* associates JESS facts with database tables in a MySQL database.

The agent cycle has three parts:

1. Get inputs of any new events in the world and receive ACL messages; these are processed by the *input_interpreter*.
2. Run the jess *FuzzyRete* engine
3. Check the JESS facts and execute any intentions which can be directly executed (for example an intention to perform a speech act); retract the intention.

8. SUMMARY AND FUTURE WORK

The work described in the paper is concerned with the design and development of 3D embodied agents which are capable of carrying out conversations with users and tailoring services to user preferences, where conversations include meaningful facial expressions and body animations. We have identified the need for interaction protocols, user modelling and an internal language for the agent's reasoning and planning. We are looking at the design of sales agents in particular, but our aim is to develop Personal Service Assistants (PSAs) which can interact with the user and converse about and personalise in multiple application domains. In order to tackle this problem we need an open agent architecture to complement the embodied agents developed on this project.

We have presented an architecture and implementation for agents situated in a virtual marketplace. We have designed a modular architecture for agents, which means that the various different agents can be built from re-usable blocks. In our implementation we have made use of the JADE platform and the JESS knowledge base and engine. We use fuzzy logic to model user preferences.

At a more technical level, the issue of believable interaction is central to our study. In order to achieve this and create the impression of intelligence we need to break away from prescripted interaction and develop an architecture, which allows emerging behaviour. Future versions of our agent architecture will include fuzzy rules in the agent's action planner so that different behaviours can be selected according to the agent's internal state or "mood".

In our account we have looked at only one protocol. An extensive set of protocols for interaction must be developed. This will be done by first making up a set of storyboards, which encompass all the various types of interactions that may occur in a particular shop. Then the appropriate protocols will be identified and specified, in the process an attempt will be made to generalise as much as possible, so that several different interactions can be encompassed by a few protocols.

9. ACKNOWLEDGMENTS

The work has been undertaken with the financial support of the SoNG project (IST-1999-10192), part of the EU-funded Information Societies Technology (IST) programme.

10. REFERENCES

[1] Charlton, P., Kamyab, K. & Fehin, P. Evaluating Explicit Models of Affective Interactions. Workshop on Communicative Agents in Intelligent Virtual Environments, Autonomous Agents (2000)

[2] Gazdar, G. Speech Act Assignment. Appearing in "Elements of Discourse Understanding" . Cambridge University Press, (1981).

[3] JADE, Java Agent DEvelopment Framework, http://sharon.cselt.it/projects/jade/

[4] JESS, the Java Expert System Shell, http://herzberg.ca.sandia.gov/jess/

[5] Lester, J., Voerman, J., Towns, S., Callaway, C. Cosmo: A Life-like Animated Pedagogical Agent with Deictic Believability. IJCAI '97 Workshop on Animated Interface Agents: Making Them Intelligent, pp. 61-69, Nagoya, Japan, August 1997.

[6] Mc Guigan, R., Delorme, P., Grimson, J., Charlton, P., Arafa, Y. 1998. The Reuse of Multimedia Objects by Software in the Kimsac System, OOIS '98.

[7] NRC FuzzyJ Toolkit, National Research Council of Canada, http://ai.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit.html

[8] Rao, A.S., Georgeff, M.P. An Abstract Architecture for Rational Agents. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR '92). Morgan Kaufmann Publishers, San Mateo, CA, USA; 1992; xv+791 pp. p.439-49. (1992)

[9] Singh, M. Agent Communication Languages: Rethinking the Principles. IEEE Computer. vol.31, no.12; p.40-7. (1998).

[10] Weizenbaum, J. ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine. Communications of the ACM. Vol. 9, no. 1; p. 35-36 (1966).