Agent Based Ontology Driven Virtual Meeting Assistant

Phil Thompson, Anne James, and Emanuel Stanciu

Distributed Systems and Modeling Research Group
Coventry University
United Kingdom
{P.Thompson,A.James,E.Stanciu}@Coventry.ac.uk

Abstract. It is widely recognized that much management time and effort is expended in attending meetings and being such a costly resource it is important that the decision to attend a meeting is not taken lightly. Because of this many busy executives are rejecting face-to-face meetings in favour of virtual meetings involving both high and low complexity technological solutions. This paper proposes an agent based, ontology driven meeting assistant designed to support virtual meetings. Situated on a graph database and operating over a peer-to-peer network the ontology will provide the basic vocabulary for the meeting and the semantics required to achieve a dialogue between the agents and the human interface used by the meeting assistant.

Keywords: Virtual meetings, Peer-to-peer, Multi-agent systems, Ontology, Graph database.

1 Introduction

This paper develops ideas explored by the author in similar papers to provide an environment to support meetings [1] using agents recognizing that meetings are becoming increasingly virtual in nature. That is to say that the meetings take place with the participants in their home locations, "meeting" their fellow participants over a network using a computer. The increase in the use of virtual meetings [2] has come about because high ranking company executives are involved more and more in meetings with global partners involving travel over large distances with the ensuing high costs [3]. It is costly not only from the standpoint of the actual air travel, accommodation and other expenses which are involved, but also because of the amount of the executives' time which is spent on route to and from meetings [4]. Even though virtual meetings are not yet considered to be more effective than face-to-face meetings one writer [5] cites the Wall Street Journal study which reported that out of 2300 business leaders 60% of the respondents used virtual facilities "very frequently". These virtual meetings are becoming more common and are being used by companies for prestigious occasions like sales conferences as well as business meetings. There are many ways for virtual meetings to be supported. These range from the simple, involving PC's and the use of software such as Microsoft Office Live Meeting, together with webcams and microphones over a broadband or similar link, to the more sophisticated

T.-h. Kim et al. (Eds.): FGIT 2010, LNCS 6485, pp. 51–62, 2010.

Telepresence [6,7] which, through the use of large screens gives the impression of all the meeting attendees being present in the same room, even though they are separated by thousands of miles. Companies have also taken advantage of Second Life, meeting other people in a virtual world disguised in the form of an avatar. [8].

It has long been recognized that in order to manage a meeting effectively a good facilitator is required [9,10,11,12]. The role of the facilitator includes: ensuring the objectives of the meeting are met and that the meeting is not going off track; allowing all the meeting participants to have input to the meeting; preventing some individuals dominating; and summarising and clarifying discussion where necessary. Clearly, if a facilitator is needed in a face-to-face meeting where all the participants are in the same room, it is more necessary in a virtual environment. In a virtual environment, the ordered scheduling of contributions into the discussion becomes much more difficult. To prevent everybody talking at once and controlling the meeting requires a lot of concentration and skill by the facilitator. Also, the fact that virtual meetings tend to be asynchronous with participants leaving and re-joining the meeting as it runs its course presents other challenges [13] compared to face-to-face meetings. For these reasons to provide computer assistance to manage the participants and to take away some of that responsibility from the facilitator, allows that individual to focus more on the content of the discussion than the mechanics.

The use of computer based agents to assist humans in the performance of tasks is a dream [14] that is still yet to achieve a "killer application" but promises to become the means by which computers can perform less important tasks while enabling humans to concentrate on more important ones.

In order to assist in the facilitation of meeting an intelligent agent requires a knowledge of the structure and vocabulary of a meeting. An ontology provides the vocabulary of words and the relationship between them, providing meaning, enabling computers to make sense of a particular domain. A graph database is a means by which ontologies can be created and maintained making it an ideal vehicle for the intelligent agent to "understand" its environment.

In order to address some of the problems mentioned above these now available technologies have been utilized in the development of a prototype "meeting assistant" designed to support the facilitator of the virtual meeting environment. This paper describes the prototype. Section 2 identifies the objectives and theoretical foundation behind the design, Section 3 gives a functional description, Section 4 describes the technical design, Section 5 explains the way the prototype has been implemented and Section 6 provides conclusions and outlines our future work.

2 Objectives and Theoretical Foundation

The objective of this paper is to report on the work to use a multi-agent approach as a possible solution to the problem of managing virtual meetings. By using an ontology to provide the means for agents to make sense of and give meaning to their environment, they can be used to assist the meeting facilitator in performing that meeting management role. This work draws on the theoretical foundation set by other researchers in these fields and attempts to introduce some new ideas to further this work. This section explains how these ideas have built on this theoretical foundation giving a brief introduction to multi-agent systems, ontologies and peer-to-peer networks.

2.1 Multi-Agent Systems

Multi-agent systems have been described as “computational systems in which several artificial "agents", which are programs, interact or work together over a communications network to perform some set of tasks jointly or to satisfy some set of goals” [15]. They are also described as either “benevolent or self-interested” in pursuit of their goals and autonomous in their choices. Although a truly independent agent should be able to choose whether or not they complete a task they would not be much use in a situation where the task is important. The particular agents that are described in this paper are benevolent rather than self-interested sharing the goals of their master (being the facilitator or the participants of the meeting) according to the requests made of them. They pursue those goals with the set of skills that they have been programmed to use. This means that they release some of their autonomy to co-operate with their master.

The Open Multi-Agent System (OMAS) [16] provides the infrastructure for the application. OMAS uses a concept of personal assistants which provides a human interface to allow a mixed environment of humans and agents in the design of applications. It provides skeleton agents to which can be added LISP code which gives the agents the “skills” required to perform the actions required by the applications. The mechanisms to allow natural language statements to be passed to agents from the personal assistant are included and these statements are then parsed to extract the “performatives” and associated information. The performatives are verbs which require the agent to execute a particular action according to its implemented skills. The creation of the dialogue and vocabulary which is necessary to make sense of the statements must also be implemented in LISP code although again skeleton procedures are provided to simplify this process. The human interface of the personal assistant provided by OMAS is suitable for prototyping the processing of these statements but what is required for actual applications is a procedure more specific to the needs of that application.

The meeting assistant described in this paper has provided an interface which is designed specifically for a virtual meeting environment and attempts to make the process of creating vocabulary and dialogue more configurable by the application designer. This is achieved by using a database which can be created and maintained without recourse to LISP code.

2.2 Ontologies

Ontologies have their origins in philosophy but work started in the 1990’s to use them to model information systems [17]. A definition given at that time was, “a specification of conceptualization”. This was further developed into, “a specification of a representational vocabulary for a shared domain of discourse – definitions of classes, relations, functions and other objects”. This has evolved today to a comprehensive specification of an ontology in the form of the Web Ontology language (OWL), Resource Description Framework (RDF) and Vocabulary Description Language (RDFS) which have become the adopted standards for applications embodying ontologies into their design. These are the standards which have been built into the ontology used in the Meeting Asssistant described in this paper. The ontology has been implemented using the ALLEGROGRAPH graph database. The ontology database has then been

used to give the agents a vocabulary for the meeting application concepts and objects and the relationships between them, to allow them to make sense of that domain. The idea of using an ontology in an application using multi-agents is not new and is a concept employed by OMAS described above. The ontology in OMAS is implemented using the LISP language which although made simpler by the use of skeleton procedures requires a working knowledge of the language to configure the information. What is different in the application described in this paper is that the ontology is embedded into a database and uses RDF statements as the means of describing the word vocabulary, object properties, relationships and command list of the application making configuration easier. The command-list on the ontology will be used to validate the keywords addressed to agents from the meeting participants.

2.3 Peer-to-Peer

In a peer-to-peer network [18] no central server is required. Each peer brings enough extra resource to support its own membership of the network. Peer-to-peer computing also allows shared information to be, "accessible by other peers directly, without passing intermediary entities. When a new member joins the team information can be sourced from any of the other members. Although this creates data redundancy it does offer data resilience by providing an answer to the single point of failure problem encountered in central server systems because no central repository is used for information. The scalability offered by peer-to-peer networks means that it is easy for new participants to join the meeting. In a conventional client-server network extra resources have to be provided at the server as more and more nodes join the network.

Peer-to-peer is used by the application in this paper to broadcast messages so that any contribution for a participant in the meeting is sent to all the participants to maintain their record of the conversation. It also allows participants to be able to communicate with each other without any central server being involved. OMAS uses peer-to-peer as the means of sending and receiving messages from one computer to another. The scalability of a peer-to-peer network allows new meeting participants to join the meeting with minimum impact on the rest of the system.

3 Functional Description of Prototype

To support the typical virtual meeting environments referred to above there needs to be both control over the dialogue between participants as well as the need to assist the facilitator in managing the meeting. To control the dialogue what is required is an orderly transfer of the dialogue from one participant to the next. Ideally it should not be possible for more than one person to contribute to the meeting at the same time and the time allowed, to give opportunity to all the participants to make a contribution should be carefully metered. The work of the facilitator is to be concentrated on bringing participants in to the discussion in turn, allowing the person who is the most knowledgeable on an issue to lead. Any offline tasks which would interfere with this process should be handled by the meeting assistant. These tasks would include such peripheral activities as issuing agendas or discussion papers to new arrivals at the meeting. This becomes especially important because the nature of the virtual meeting

is asynchronous, with participants joining and leaving the meeting more so than happens in a synchronous or face-to-face meeting. The usual expectation in the face-to-face meeting is that people arrive at the beginning, receive all the supporting documentation, take part in the meeting and then leave at the end. A virtual meeting is likely to last longer, sometimes spread over many days and people would be more selective over the agenda items to which they contributed.

Using a suitable input device, the participant will, on registering for a meeting, be able to access any information relating to previous related meetings including a record of the discussion, summary of agenda items and actions given to individuals together with any supporting documentation. On requesting to join the meeting the participant will receive a list of all the meeting participants together with the agenda for the meeting and a summary of the meeting so far if the meeting was in progress. The participant will have a visual display of the typed contributions of all the other meeting attendees as they occur and will be able to make a contribution if required. All requests to contribute to the meeting will be directed to the facilitator who will cue the participant when it is their turn. The participant will inform the facilitator if they wish to leave the meeting which could occur through interrupts at their location or simply because they have no interest in the particular agenda item. This will remove that individual from the list of current attendees.

The facilitator will move through the agenda items in turn and after enough discussion has taken place will summarise the main decisions, record any actions against individuals together with the date the action should be completed and then close the agenda item. If it is judged that there has not been enough discussion on an item or because information is needed from somebody not present at the meeting the facilitator may choose to leave the agenda item open and proceed with the meeting. When all the agenda items have been discussed the facilitator will either suspend or close the meeting depending on whether any agenda items are still open. The facilitator may choose to interrupt the meeting or the contribution of a participant at any time. If the meeting is interrupted in this way all participants will be notified of the reason and informed when the meeting will continue. This could happen if the facilitator needed a break in the proceedings or wished to suspend an agenda item to bring a more important item forward for discussion or schedule in a previous agenda item left open because a key individual has now joined the meeting. Another option open to the facilitator particularly if there is a need to attend to some other business, would be to pass the facilitator responsibility for the meeting to another individual. In practice any one of the meeting participants could take control of the meeting simply by giving them access to the functions normally given to the facilitator. A third option would be to allow the meeting to run on “automatic” at least as long as there are still requests to contribute from participants when the meeting would be temporarily suspended until the return of the facilitator. This would mean that there would be no supervision of the dialogue with the atttendant risk of the discussion moving “off topic” and open to abuse.

When agenda items require formal agreement by the meeting participants they will be able to propose or second a motion and then agree, disagree or abstain in any voting activity. This will be recorded against the agenda item for future reference. Voting preferences will not be recorded against any individual in the meeting apart from indicating whether they have voted or not. This will allow votes to be cast by individuals later before the agenda item is closed.

The ontology will contain a vocabulary of the terminology used in the meeting and any relationships between words or concepts within that vocabulary. So for example the ontology will "understand" that the "agenda-item" relates to an "agenda" which itself relates to a "meeting" and so on. Also contained in the ontology will be a series of "performatives" which will allow the participants to issue natural language statements to the meeting assistant which will be recognized by matching against the ontology and result in certain actions being performed by the assistant. For example the use of the performative "get" together with the word "agenda" will result in the meeting agenda being sent to the participant. Properties will be stored against these terms to help the meeting assistant decide the action required. For example against the "agenda-item" will be a filename or URL to allow the information to be accessed from a database or on the web. The ontology will be stored on a database to allow easy updating of the terminology, relationships and properties for different types of meeting.

For efficient management of the meeting there needs to be a facilitator. The facilitator needs to create an agenda where the agenda item sequence is clear for all participants and an estimated time for discussion for each item should be added. Ideally somebody should be asked to lead the discussion on each agenda item. At the end of each agenda item the facilitator should summarise findings and then clearly identify what actions are required and who is to perform the action [14]. A date should be associated with action to ensure feedback.

The participants on joining the meeting should be given a copy of the agenda and any other supporting documents. The facilitator should control who speaks and when they speak and should be able to interrupt if required to seek clarification. Participants should be able to signal to the facilitator to gain access to the meeting; when they wish to speak; if they do not understand; if they want to move progress; if they wish to leave the meeting, etc. They should be able to terminate a session and resume at a later time in case they do not need to be present for certain agenda items. At the end of the meeting they should be provided with a copy of the action list by the facilitator.

4 Technical Design of Prototype

The prototype environment we designed consists of control windows for the participant and facilitator. These control windows are used to display the discussion and allow communication and also provide access to the infrastructure which supports the meeting. The environment is designed to allow all communication between participants, facilitator and the meeting assistant agent to be controlled from the one window using a conversational dialogue.

The Input-Output window in Figure 1 is the first window to appear after login. It gives the participant a list of all the other meeting attendees by displaying their names down the right hand side of the screen. The discussion itself appears in the main part of the window with a scrolling record of the contributions made showing the name of the participant together with their typed contribution. Participants wishing to contribute to the meeting discussion will type their contribution and then hit the "send" button which will display the entry in the scrolling record after the entry of the current participant has been completed.

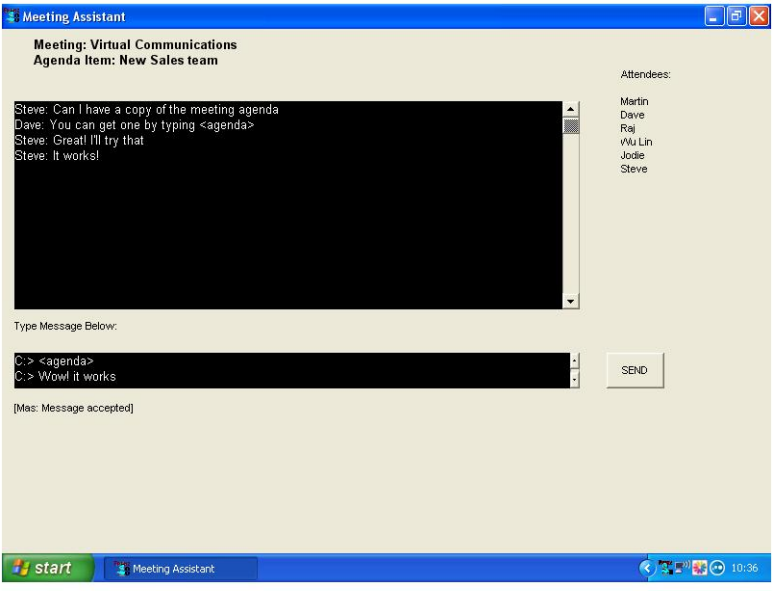

Fig. 1. Input-Output Window

The participant can also communicate to others in the meeting by highlighting them before hitting the send button which will direct the message to those selected. This facility can be allowed or disallowed at the discretion of the meeting facilitator. It could be useful to allow the facility if discussion is required between members before a decision can be taken. Messages received back will only appear on the screens of the selected participants and the original sender.

Participants will obtain information about the meeting by the use of keywords they type into the message. These messages will be targeted at the meeting assistant which will process them accordingly. The list of keywords and how to use them will be sent to the participant on successful login. They will include: get agenda, get agenda item, list attendees, etc. Any information sent in response to these messages will be displayed in pop-up windows on the requestor's main screen.

The facilitator (who could be at the same time a meeting participant) will also use the display and communication window. Using that window, agenda items will be introduced; the flow of the meeting will be interrupted when required; the agreed actions will be typed when all the discussion on an agenda item has been completed and the meeting will be closed. All this will be visible to the other meeting participants in the scrolling display, which will become a permanent record of the meeting.

All the messages from participants will be directed to the meeting assistant which is an intelligent agent who through a system of keywords embedded in the messages will be able to perform the necessary processing of the message. Keywords identifying the sender and target of a message will precede and follow the message and the agent will direct the message accordingly, messages being displayed on the scrolling record in the display and communication window of the target participant. Normal messages not containing a target keyword will be directed to all participants.

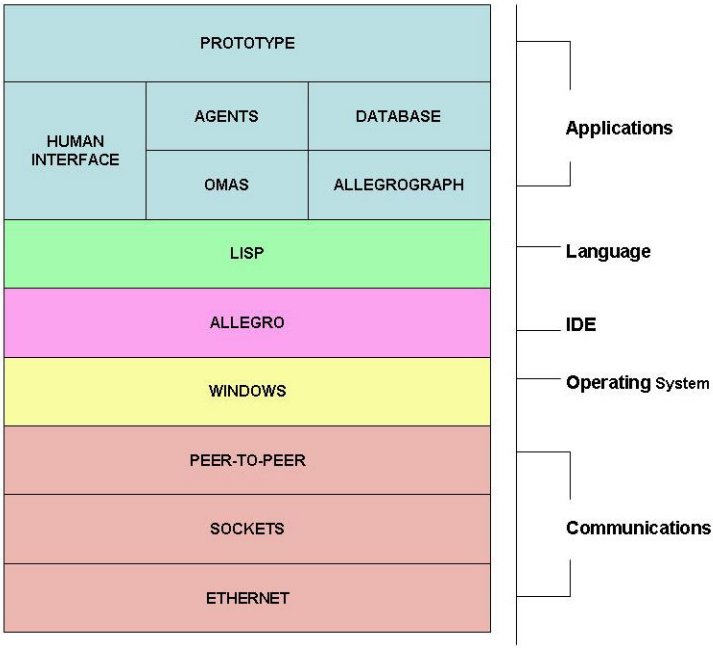

Fig. 2. Prototype Implementation Layers

Where the message target keyword specifically identifies the meeting assistant this will indicate the requestor requires information about the meeting which will necessitate sending messages to other agents called meeting agents who will process the message and obtain the required information. The meeting agents will use the other keywords in the message to access the meeting ontology and extract the required information to be sent back to the participants.

5 Implementation of Prototype

The implementation layers of the prototype are shown in Figure 2. The application layer of the prototype is made up of the human interface, the agents and the database. The human interface is where the human participants and facilitator interact with the system. The agents, consisting of the meeting assistant and meeting agents operate within the OMAS environment and interact with the human interface by sending and receiving messages. The graph database using the ALLEGROGRAPH package is updated by the meeting agents and holds the meeting structure, vocabulary and list of commands in the form of an ontology.

The applications have been developed in the LISP language using ALLEGRO as the Interactive Development Environment (IDE) and operate under the control of the WINDOWS operating system.

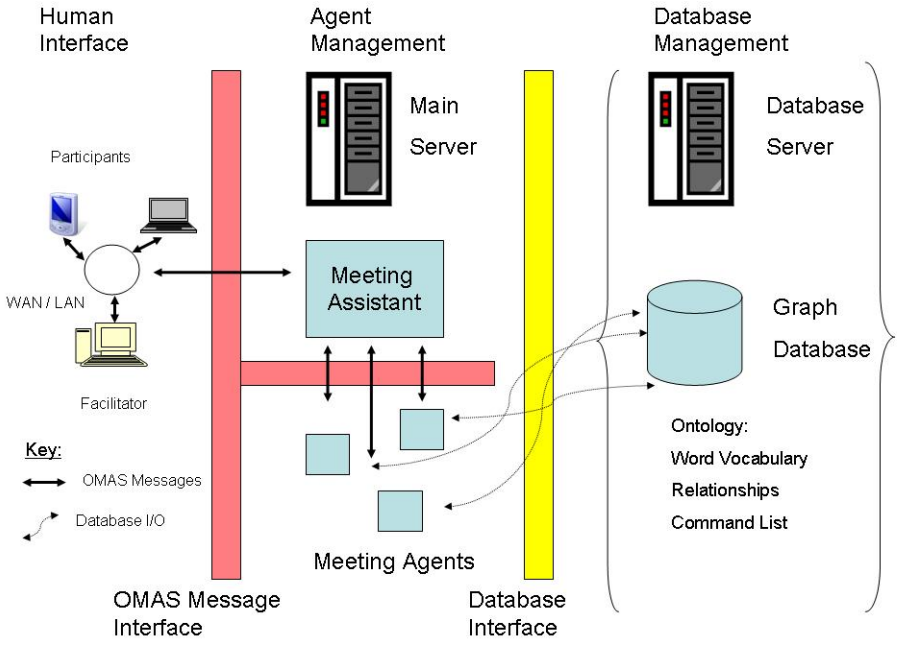

Fig. 3. The Protoype Design

The communications layer controls the interaction between the different nodes on the Ethernet network. The facilitator and the participants each occupy a node on the peer-to-peer network. The communications between nodes is implemented by the use of sockets which identify the IP address and port number of each node. The same mechanism is used to communicate with the agent infrastructure through the use of messages being sent via the sockets to and from the human interface and the agent infrastructure.

The OMAS agents process the messages according to their implemented "skills" in order to service the requests coming from both the facilitator and the participants. The prototype is shown diagrammatically in Figure 3 which also illustrates how the components of the virtual meeting environment can be located on different computers.

The meeting ontology is achieved through the use of a graph database which, through the use of a triples datastore, identifies the terminology used in the meeting environment, the relationship between objects reflected in those words and the properties of the objects. An extract of the ontology for the meeting is shown in graph form in Figure 4, and in the Resource Description Framework (RDF) in Figure 5 which illustrates how this information is held in the database in triples format. The equivalent graph is shown for the Command List in Figure 6.

When the meeting assistant recognizes a command keyword in the input line from the participant the command is extracted and sent to one of the meeting agents. The meeting agent first validates the command performative against the graph database returning an error if it does not match the database. If the performative is valid the agent processes the rest of the command and by breaking it up into recognizable

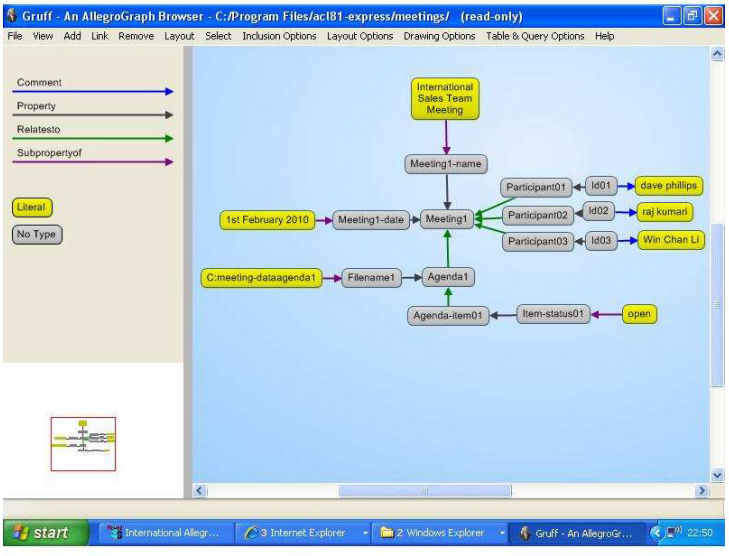

Fig. 4. Meeting Ontology Graph

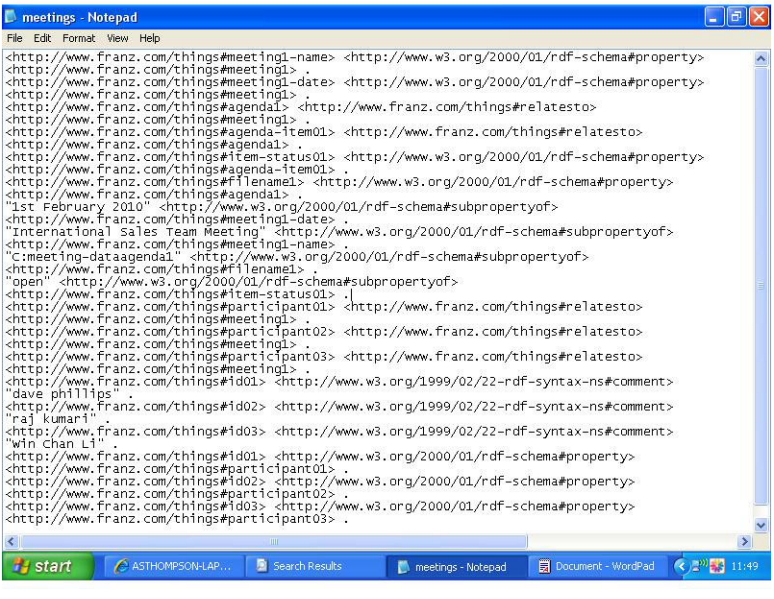

Fig. 5. Meeting Ontology Graph – Vocabulary

elements uses those elements to perform the processing. The result of the command could update the database with information passed from the participant or extract information from the database to be sent back to the participant. The processing of the database is performed by building up LISP queries from code fragments held on the database according to the type of processing required.

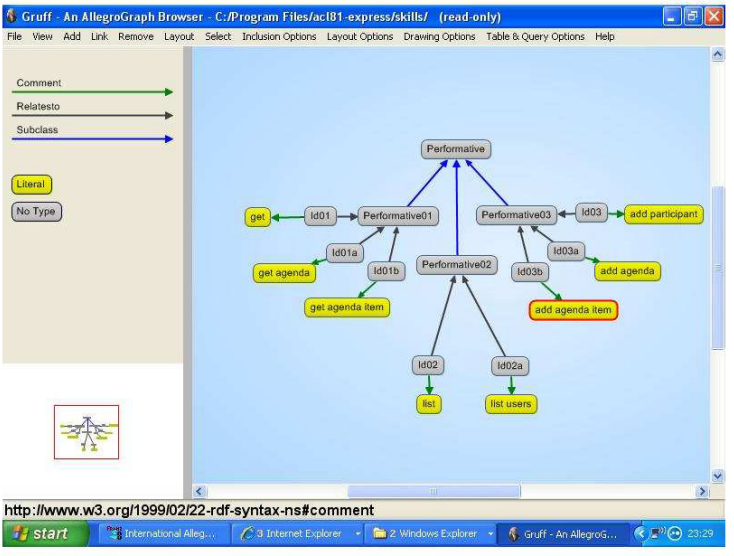

Fig. 6. Meeting Ontology Graph - Commands

6 Conclusions and Future Work

Techniques and solutions have been referred to in this paper which use computing technology for supporting virtual meetings and this is now an established option for business. However, the novelty of the system described in this paper is that the natural language dialogue from the meeting participants and facilitator is interpreted by the use of an ontology and given meaning to allow requests to be performed by "intelligent agents" in response. It provides the means to manage virtual meetings by allowing the participants to see the contributions of the other participants, add their own contributions and also to obtain supporting information and manage the agenda through a simple natural language interface. The facilitator can concentrate on the important task of co-ordinating the agenda and the contributions of participants while being assisted by multiple computerized agents with the less important tasks. The structure and vocabulary of the meeting understood by the agents is implemented as an ontology which, residing on a database can be easily created and maintained.

Future work will be concentrated on further refinement, trialling the application in a real meeting situation, analysing the results and reporting the outcomes.

References

1. Thompson, P., Iqbal, R.: Supporting the Social Dynamics of Meetings using Peer-to-peer Architecture. In: Proceedings of the 15th International workshops on Conceptual Structure, CSCWD 2007, pp. 170–176. Springer, Heidelberg (2007)

2. Thompson, P., Iqbal, R., James, A.: Supporting collaborative virtual meetings using multi-agent systems, cscwd. In: Proceedings of 13th International Conference on Computer Supported Cooperative Work in Design, pp. 276–281 (2009)
3. Armfield, R.: Virtual meetings save real money, Bank Technology News 23(7), 13 (2010) (AN52411030)
4. Ellis, C., Barthelmess, P.: The Neem Dram. In: Proceeding of the 23rd Conference on Diversity in Computing (TAPIA 2003), pp. 23–29. ACM Press, New York (2003)
5. Boehmer, J.: Harvard study shows face-to-face meeting value, rising virtual interest. Meeting News 33(12), 9, 1p, 1 Chart (2009)
6. Scofidio, B.: Why Should(n't) You Manage Virtual Meetings. Corporate Meetings & Incentives 27(8), 4, 1 (2008)
7. Bulkeley, W.: Better Virtual Meetings. Wall Street Journal-Eastern Edition 248(75), B1-B5 (1987)
8. Kharif, O.: The Virtual Meeting Room. Business Week Online, 6, 1 (2007) AN 24781227
9. Black, A.C.: Getting The Best From Virtual Meetings. Bloomsbury Business Library – Manage Meetings Positively, 60–71, 12 (2006)
10. Miranda, M., Bostrom, P.: Meeting facilitation: process versus content interventions. Journal of Management information systems 15(4), 89–114 (1999)
11. Barker, A.: Crash Course in having effective meetings. Management Today 22(2/3) (June 2008) (AN 32828858)
12. Scofidio, B.: People Management.: How to have effective meetings, vol. 14(20), p. 45, 1 (2008)
13. McQuaid, M., et al.: Tools for distributed facilitation. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, p. 10 (2000)
14. Fisher, L.: Make way for intelligent agents, cited by: Strategy & Business. In: Booz & Company 2010 (1995)
15. Lesser, V.: Encyclopedia of Computer Science, 4th edn., pp. 1194–1196 (2003)
16. Barthes, J.-P.A.: OMAS – A flexible multi-agent environment for CSWD. In: Computer Supported Co-Operative Work in Design 2009, pp. 258–263 (2009)
17. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993)
18. Kellerer, W.: Dienstarchitekturen in der Telekommunikation – Evolution, Methoden und-Vergleich. Technical Report TUM-LKN-TR-9801 (1998)