Introducing Participative Personal Assistant Teams in Negotiation Support Systems

Eric Platon and Shinichi Honiden

National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan
{platon,honiden}@nii.ac.jp

Abstract. This paper introduces teams of personal agents that support users individually in electronic negotiations. These agents listen to the running negotiation and to each other to point out relevant information and compile advice for the user. In this frame, we first describe the architecture of this system and propose assistance interaction protocols to specify agent external behaviours in performing their tasks. Then, we discuss the semantic representation of agent communication and describe an abstraction layer to let agents understand user message issues. Our future work aims at improving these mechanisms and enriching them toward a full-fledged implementation.

1 Introduction

The last decade reveals a multiplication of software agents that organize our time, advice in booking airplane tickets, perform auctions on our behalf, or maintain business process [1–4]. In most systems, a central mechanism is the negotiation between cooperative or conflicting agents that need ways to reach an agreement in the fulfilment of their tasks. Most projects also have in common to assign a single agent to each user and to concentrate on the challenging concept of delegated negotiation in which agents autonomously act on behalf of their owners, under customised constraints. The restriction to one agent is natural as users need to deal with one entity at a time, but it seems also limiting when comparing standalone agents to multi-agent systems (MAS). We argue that teams of specialised agents can ease both the understanding of the system behaviours for the user and the engineering of smaller interactive software for the system designer. Delegated negotiation is a very active discipline in multi-agent systems [3, 5, 6], but we think potential users are still reluctant to delegate any power to artificial agents in affairs concerning personal issues. Assistants that act *with* the user should be more easily trusted, since they mainly suggest possibly relevant information or anticipate user needs rather than negotiate automatically.

In this context, we introduce a *personal assistance system* (PAS) that features assistant teams to participate to negotiation processes aside the user, rather than on her behalf. This intermediate stance between manual mode (no support) and delegated negotiation should steadily strengthen user trust in artificial assistance, and probably lead to higher acceptance in delegation. Such PAS have

M.W. Barley and N. Kasabov (Eds.): PRIMA 2004, LNAI 3371, pp. 178–192, 2005.

already been partially explored with single assistants [2, 7], but we will describe active teams that clarify the system reactions to the user, and permit flexible management of the agent population in terms of available service (user point of view) and software engineering (incremental population of more simple units).

The paper is organised as follows: we first situate and motivate our research in the field of Negotiation Support Systems (NSS) in section 2. After presenting our settings and assistant teams in section 3, we introduce in section 4 our methodology describing team protocols and participative features that address the system issues. In section 5, we detail work related to NSS and assistance, and finally compile our current concerns and future work in section 6.

2 Negotiation Support and Participative Assistance

This section is first devoted to the description of our concept of PAS relative to NSS and the motivations to propose another model for assisted negotiation. Then, we highlight the issues we address in conducting this research.

2.1 Personal and Participative Assistance in Negotiation

Online negotiation becomes a standard so that NSS appear in numerous projects. For most researchers, agent-mediated solutions represent appropriate models, mainly for the challenging delegated negotiation. However, user reluctance stems from the idea of transferring decision power to artificial agents, since delegation-based software usually do not reach the 'trust threshold', as shown on Fig. 1. We think this threshold is shifting and people will accept such a support in the future, when solutions are robust enough to deaden most worries about automated processes. The work of Klein et al. advances in this direction [8], and steady introductions in the industry should encourage for acceptance.

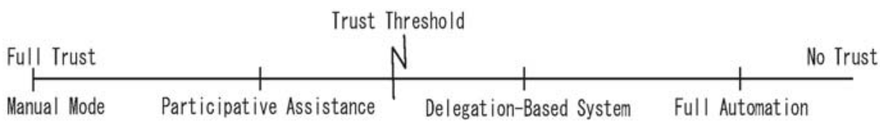

Fig. 1. Our distribution of participation and delegation on the trust ladder.

Meanwhile, we suggest an intermediate stance to address this user reluctance. PAS are NSS that negotiate *aside* rather than *for* users, so that it solves intrinsically the delegation concern. We reuse the term and concept of *participation* exploited by Drogoul in Agent-Oriented Simulation [9] to refer to our type of negotiation assistance, where user and assistants are paired and bound with the same aim. If this approach is accepted by users, we think it may reduce the gap for eventually accepting delegated negotiation. Little work already exists [7], and the proposal we introduce should lead farther. Indeed, our vision of assistant is strongly akin to human one. For example, a secretary initially executes exactly chief's orders, provides few valuable feedbacks, and has no power.

After bilateral adaptation, the chief increasingly trusts the secretary and delegates more power; the secretary learns about chief's methods and can anticipate some requests and give relevant information. Hence, some secretaries negotiate efficiently the real business schedule of chiefs that trust them. Thus, PAS aim at designing assistants that reply to user requests, and also take such meaningful and understandable initiatives that can serve the user in its activity. The latter functions include searching non-requested information related to the context, suggesting alternatives, and so forth.

Second, NSS solutions are restricted to one single assistant. Standalone agents can become fairly monolithic in this situation and cannot leverage the potential of MAS. One central requirement from users to accept an artificial agent is to sufficiently understand and predict its behaviours. In complex knowledge-based systems, explanation facilities are exploited, but we think a team of simpler agents can generate easier justifications of their actions, in addition to explain their interactions. Moreover, the engineering of simple agents that cooperate allows dividing design over specialised entities and their interaction patterns, i.e. a multi-agent system.

Beyond the case of negotiation assistance, we finally see personal agent teams as the future of user interfaces. Trends like ubiquitous computing show software belongs to our private daily life through our mobile phones or PDA. One day users might head a family of persistent agents that assist them in their digital life. A specialised team like our proposal can lead to an appropriate foundation.

2.2 Research Issues

The purpose of this work is first to design adequate interaction protocols that orchestrate agents to reply to user requests, and enact the initiative feature of efficient assistants. These protocols must deal with both the assistance provided by one single agent (e.g. simple smart search on Internet) and a group of cooperative agents (e.g. the search agent initiates a search on its own to assist another agent – itself assisting the user) to provide advanced services and straightforward identification of actors by users.

Once protocols set up the required infrastructure, assistance needs representing and reasoning about events and their semantics. This provides an appropriate knowledge of the current affairs and allows agents reacting accordingly. In addition to simply communicate directly as in most current MAS, we think agents should be endowed with mechanisms to listen to others' interactions. Indeed, the recent concept in MAS named *overhearing* [10, 11] provides much more knowledge resources that agents can exploit to improve their services.

Such mechanisms emphasize the issue of communicative act semantics with assistants. Natural Language Processing (NLP) is a very active field, but free communication with computers remains a technical challenge. Computer-based assistants suffer from the poor meaning that we can currently embed in user interfaces. An abstraction layer for communication should normalise the form and semantics of messages input to the system, in a specialised and restricted way of the Semantic Web [12]. Such an alternative could convey human- and machine-understandable meaning among assistants and users. In our context, it should be a compromise between *simplicity* to bypass most NLP challenges and *richness* for our purpose of negotiation support.

Finally, our future view of daily life assistants requires flexibility for users and software designers. One may want a dynamic population of communicative agents to add or remove services, upgrade or customise heterogeneous components. This is directly related to an incremental design approach that allows such dynamism and also reduces the complexity of each piece of agent. This flexibility at runtime represents one major reason to consider MAS. In the scope of this paper, we describe PAS as a solution to address those above issues, the least advanced state being the knowledge management part.

3 Assistant Team Model

3.1 Architecture

The central mechanism of our negotiation assistance model is the interaction among agents and humans. Our infrastructure is laid out on Fig. 2 among three negotiation participants, together with their assistant teams.

Conversely to most NSS (see Sect.5), agents do not substitute users but participate to the overall process discussing, listening to exchanged messages, and reacting for their owner's sake. Rather than delegating the negotiation performance to their agents, users act directly with other parties and keep full control of their strategy. Assistants stay aside and intervene as necessary on

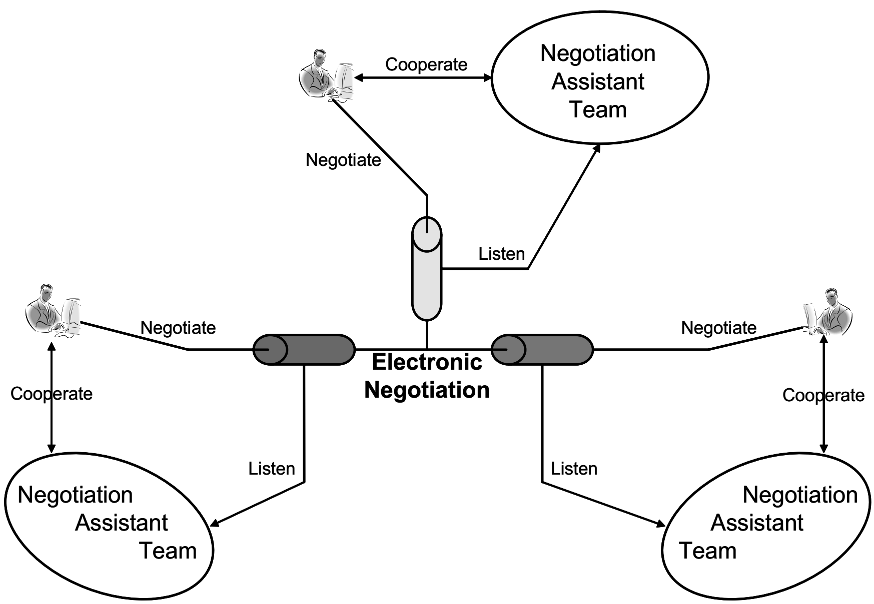

Fig. 2. Architecture view: online negotiation and assistant teams.

user invocation or key event occurrence (detection of irregularities, discovery of relevant information or even some comparison results with past events).

Each user is endowed with a *team* of assistants, i.e. a set of agents that collaborate to provide support. Services first consist in individual activities of each assistant (such as search, history, or strategy advice) that produce local arguments for the system and perform tasks directly required by the user. Second, the service features a system-level argumentation synthesising individual grounds to deduce a global argument. The user can access both individual and global information depending on the strategy (one may prefer peculiar types of data) and the will to trust system conclusions. The main idea is to consider this team as a board of advisors in a meeting room. All participants discuss the agenda under ruling of the chairman, who is the user in our present case, and they provide personal opinions to allow the chairman synthesising all viewpoints to define the company's strategy.

3.2 Illustrative Example

This section describes a negotiation assistant team through an example. Our illustrative scenario is stated as follow: John in Australia and Takezo in Japan have decided to meet in front of Victoria station, London on their common free day and agree upon the details by email. The email client has an interface that allows assistant teams performing their service. They both come by different airports. *The negotiated issue is to decide the meeting time.*

The agent team that intervenes in this example features three different specialists:

- *Presentation Agent* manages the user interface (proxy), interprets user events as communication acts, broadcasts events, and compiles global arguments.
- *Search Agent* can access the Internet to feed agent peers with fresh and parsed information, and can initiate complementary search on its own.
- *Strategic Agent* computes information and events under the essential negotiation point of view and breeds strategic arguments.

First, users negotiate along with their required Presentation Agent and one single Search Agent. We then run the scenario once more, introducing a strategic agent dynamically to the assistance population. The Presentation Agent plays the unique role to bridge the actual communication between user and assistants, albeit each agent virtually interacts with all peers and the user. When other agents 'talk' to the user, they indeed talk to the Presentation Agent. This allows treating all interactions similarly. Initial beliefs of our agents are stated in Table 1.

The first run starts with this initial knowledge when John writes a proposal:

I will be at Heathrow at 8am, so let's meet at lunch time. (1)

Before relaying the message to Takezo, the Presentation Agent (P) broadcasts it in the system. The Search Assistant (SE) knows from its beliefs the context of meeting organisation and switches to the schedule issue. Typically,

Table 1. Initial Agent Beliefs.

<table>
<tr><th></th><th>John's Assistants</th><th>Takezo's Assistants</th></tr>
<tr><td rowspan="2">Search Agent</td><td colspan="2">1. Internet time table search engine
2. In meeting issues, users usually need time table information
3. Meeting place: Victoria Station</td></tr>
<tr><td>4. User schedule (Heathrow 8am)</td><td>4'. User schedule (Gatwick 10am)</td></tr>
<tr><td rowspan="2">Strategic Agent</td><td colspan="2">1. User intention to organise a meeting
2. Check inconsistencies in schedules (overlapping, etc.)
3. Meeting place: Victoria Station</td></tr>
<tr><td>4. User schedule (Heathrow 8am)</td><td>4'. User schedule (Gatwick 10am)</td></tr>
</table>

it autonomously checks online time tables for transportation means and times along John's itinerary. John can ask for all the results directly to this assistant and require farther search. On its own initiative, SE can also inform John about any relevant result it may find (strikes, track engineering, etc.). A timeout ends the system deliberation if nothing occurs, then P can relay the message. At this point, John's assistants have no belief regarding Takezo for the meeting time question.

When Takezo receives the message, the Presentation Agent listens to this virtual discussion and broadcasts the incoming information. While Takezo is considering the new elements, assistants update their beliefs, infer intermediate results and verify the applicability. For instance, SE gets information to reach Victoria station at noon and suggest its results automatically. Takezo can exploit this behaviour and answer back to accept, deny or propose an alternative.

The second situation introduces a strategic agent in the assistant population. This can be a new run of the above scenario or users can dynamically activate the new service in the system. From the same first proposal sent by John, P and SE have the same reaction. The Strategic Assistant (ST) infers from (1) that John must be at Victoria station by noon, so it checks that John is free on this period and can reach the meeting point on time. The former check is immediate as the schedule is empty. For the second check, ST needs search support. It forwards the verification request argument to SE and waits for its reply. John can consult ST to check its trace of reasoning and perhaps find inconsistencies in the offer (1). In the case where SE replies in a timely fashion (the timeout managed by P ensures system reactivity for the user), ST completes its concern verification and validates John's idea as there is no apparent reason to fail, so that P eventually sends the proposal.

On the side of Takezo, the same analysis is performed by P and SE. ST also listens to the discussion and to SE, since the critical strategic point is the schedule here. ST deduces from its beliefs and SE results that the schedule is a little tight. Takezo can consult each agent trace or their compiled advice, and can reply:

I am not sure to reach Victoria Station by noon. How about 1pm? (2)

Before relaying this message, assistants process it as stated for John and the negotiation process continues.

From this basic example, we intend to build a system that address general negotiation situations including buying, organising time and solving conflicts. It appears from the example that interactions between assistants imply part of the system 'intelligence' and performance, while the incremental and dynamic agent population serves both users and software engineers.

4 Protocols and Participation Methodology

In this section we first present our models of assistance protocols to orchestrate the system. Then, interactions carry messages that specialise assistance in negotiation, so we describe how agents participate by exploiting message semantics.

4.1 Interaction Protocols

The assistant team should behave as a coherent whole at the user level. Agents in teams have the common goal to individually and collectively serve the owner by local and global interventions that can be described by interaction protocols. In our research, we aim at protocols that verify properties related to the concept of participative personal assistance. First, protocols must depict interactions representing both user services and cooperation among agents. This last feature details how specialised agents build up arguments they cannot produce independently, and consequently better serve the user while their actions remain understandable and dynamic. Second, the agent population must be freely decided by the user so that protocols should be as much as possible independent from the number of actors (our proposal do not currently fulfil this point). Finally, protocols must enforce agent reactivity with time management. Indeed, the agent debate is meaningful to the user only when available in a timely fashion. For instance, we address this issue in our protocol by endowing the Presentation Agent with a timeout. All system agents must submit their results on time, if relevant. This constraint binds the reasoning cycles, so that users receive support on time at the cost of shallower performance. Although most simple agents can complete their individual tasks on time, crossing and revising arguments can be time-consuming. Time bounds mostly limit the revision depth. Experimental results to explore this claim are required and consist in part of our future endeavours.

From these properties, we split interaction protocols into two connected parts. First, the *assistance diagram* presented on Fig. 3 intends to formalise with the FIPA syntax [13, 14] how assistants and users are situated. This first schema refers to the second part that describes the *assistant team diagram* deployed on Fig. 4 in the case of our example. It is the effective interaction protocol between assistants. Messages exchanged in these diagrams do not follow exactly the FIPA Communicative Acts recommendation [14]; we chose mnemonics instead of the FIPA language for space and readability reasons. For example, 'invoke' message refers to the 'request' in FIPA and could be defined as in Code 1.

Fig. 3 shows the Presentation Agent as user proxy, i.e. a relay interface with other assistants. The user may invoke the system intentionally (Alternative 1;

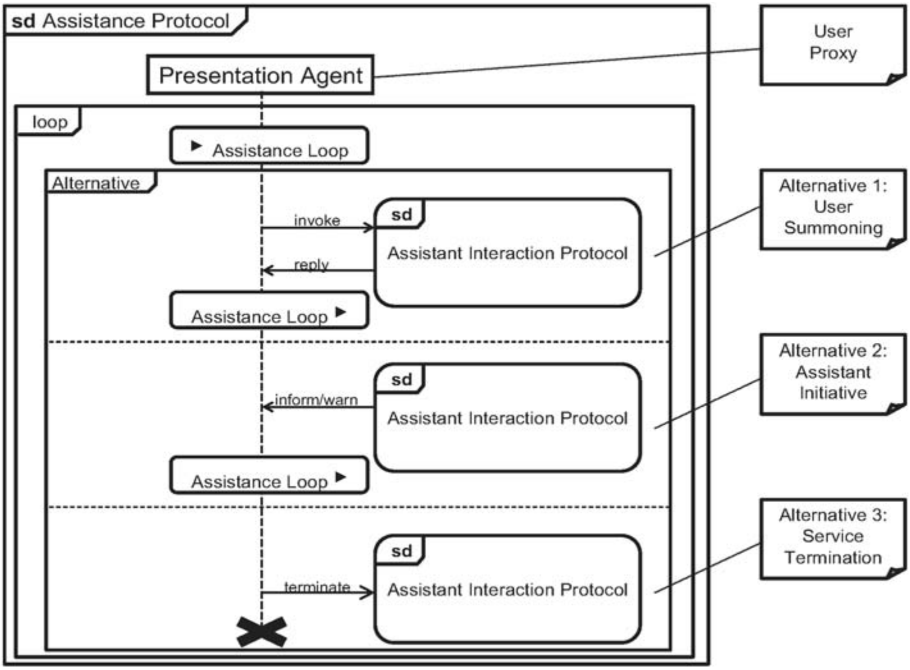

Fig. 3. Assistance Protocol: the Presentation Agent is the user proxy.

e.g. consult the Search Agent), trigger its invocation by issuing an event (Alternative 1: send a message to other parties), or terminate the agent service (Alternative 3). Furthermore, real assistants would take initiatives (inform about sudden bad weather conditions, strikes, etc.) and this is modelled with Alternative 2 in the diagram. Invocation and agent initiative activate the external protocol 'Assistant Interaction Protocol', partially laid out on Fig. 4.

Code 1. FIPA 'request' to formalise the message 'invoke'.

```
(request
    :sender (agent-identifier :name PresentationAgent)
    :receiver (set (agent-identifier :name SearchAgent))
    :content
        ''(search list route A B)''
    :language lisp)
```

The FIPA recommendation reuses the gates from UML2.0 to connect two diagrams. As the standard has no explicit representation of the gate, we present it on Fig. 4 similarly to an agent and its lifeline as it can be confounded with the Presentation Agent. Cases presented on this schema are limited to the 'invoke' and 'terminate' messages from P. In the global invocation process, agents act in

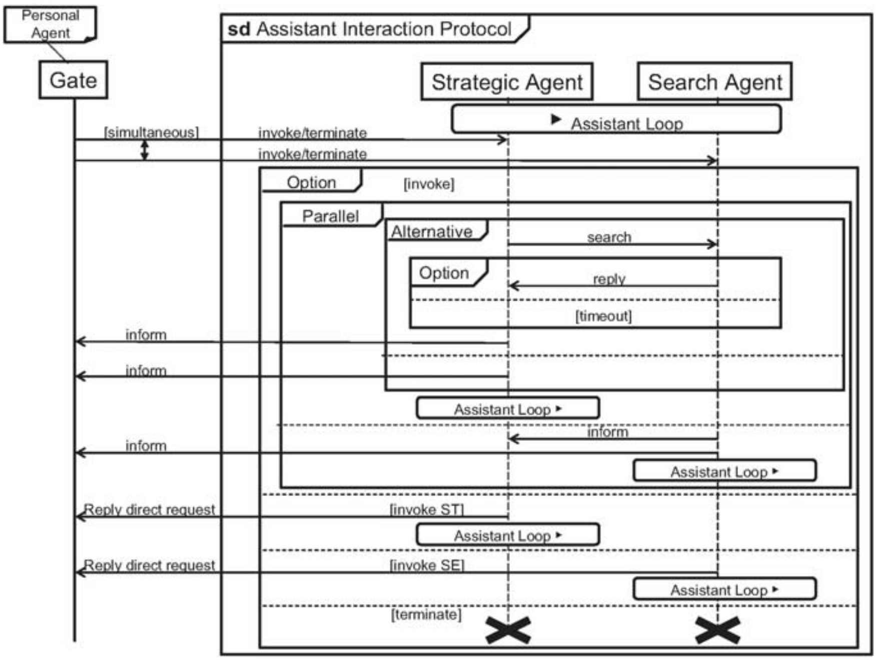

Fig. 4. Assistant Interaction Protocol, Partial View limited to user invocation.

parallel, as described in our example scenario. If the invocation targets only one agent, other agents listen to the event and only intervene in case they can infer critical arguments.

When the user sends a message to another one (initial invocation on top of Fig. 4), the Strategic Agent (ST) may ask for a search to the Search Agent (SE), and wait for an answer before informing P about its results (possibly after revision exploiting SE information). Otherwise, ST may already hold sufficient results and directly contact P. Concurrently, SE can inform on its own initiative ST about relevant events (strikes, etc.) and reports to P. The individual invocations are limited here to reply to user's requests. Extensions of our model will allow the polled agent to send messages to peers in this case. Finally, the termination case ends the assistant protocol and asks for shutdown.

The assistant diagram is designed to be a generic assistance model. Our current status for the assistant team diagram is still idiosyncratic and our endeavour is to deduce a generic protocol that still validates the previous properties, independently of the agent population.

4.2 Assistance and Participation

Besides interaction protocols, our system requires semantic process to carry on user assistance. The semantics is expressed at two levels, namely the interaction patterns and the message content carried by these interactions. Although our illustrative scenario was detailed with natural-language syntax for readability, formal communication and argumentation are needed, such as the typical illustrations by Parsons, Ramchurn [15, 16]. Implementation of these semantics enables constructing automatically sound, pragmatic, and both user- and machine-understandable arguments.

First, interaction patterns may carry meaningful information about the behaviour and intention of agents, such as cases where one can infer that running interactions follow the contract net protocol or an intimidation process. Hence, our agents should identify some recurrent or unusual patterns in the user-user, user-agent, and agent-agent discussions. The method of Sabouret allows one agent extracting knowledge from its interaction patterns with peers by compiling and integrating relevant information in *chronicles* [17]. These time-dependent internal representations contain summaries of interactions by grouping similar events in behaviours. Regular and essential features of patterns are maintained so that the agent can reason about them, react in accordance, and explain their occurence. Agents can answer common sense questions such as ‘why do you turn left?’ or ‘what is on the table?’. Our model will extend this method so that agents are furthermore able to process knowledge from any multi-agent interaction patterns they can listen or observe, say ‘overhear’, as in the simplest case depicted on Fig. 5. The mechanisms rely on the assumption that listening agents receive ‘copies’ of interaction messages so that they can integrate them and intervene when they consider it necessary and on time. The concept of assistant team is here a foundamental requirement as these copies imply cooperative agents.

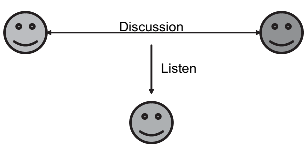

Fig. 5. Listener Agent.

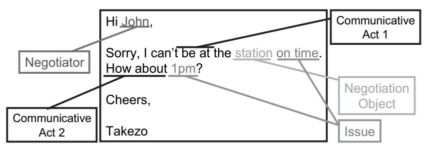

Fig. 6. Semantic Mapping.

Second, the participation of agents to the user activity needs one more important brick at the interface level to exploit message content. From well-defined semantic languages and ontology, one can build user-readable arguments automatically. However, the converse action to understand user input is a stumbling block. Our abstraction layer to address this issue aims at appropriate compromise between expressive power and complexity alleviation by well-formed arguments mapping. It is mostly based on a MAS version of the project in [18] that exploits Natural Language packages. The Presentation Agent (P) filters all messages issued and received by the user and builds up with her a temporary ontology of the running negotiation. In the case of multiple negotiation threads, the assistant selects the corresponding ontology by extracting case data from the message (who, what, etc.). The ontology serves to write down standard FIPA-ACL messages so that all agents can acquire the appropriate knowledge. This process starts with the negotiation initiation. The user sends or receives a natural language message from a peer negotiator. P parses this message and extracts term candidates for the negotiation context (communicative acts, who, what, issues, etc.) in the local ontology. These candidates are then validated by the user so that P is sure the semantic mapping is correct. To reduce the inconvenience of this participative process, we think a proper presentation to the user is required. Thus, the original message is shown to the user with highlighted term candidates and semantic annotation. Figure 6 lays out this interface. The user can click on incorrect mappings and specify the right ones. Once the agent understanding is ensured, P can compile its ontology for reuse with next messages (learning stage), translate the message into ACL, forward the original message to the recipient, and the ACL message to system agents. For a given process, P is increasingly efficient as it learns the right semantics and requires fewer user interruptions (like the secretary we introduced earlier). This method maintains sufficient expressive power for our case and bypasses the language processing barriers. It however lacks convenience for humans and need the definition of a rough natural language parsing.

5 Related Work

This section comments work akin to assistance systems and current NSS technology in comparison with our assistant team.

5.1 Assistance Projects

The Helper Agent (HA) from Isbister et al. [19] has similarities with the model we presented in this paper. In computer-supported communication, instant messaging is very popular and HA aims at improving and motivating its usage to better connect people. The main differences with PAS are that HA is standalone middleware that communicates with all actors of the communication channel, and is designed in the context of friendly relationship between actors. Our assistants form a *personal* multi-agent system that supports only one user, and will interact with outside agents in the future. So there is no assumption about the relationship between human users. Our example with John and Takezo would be similar if they are friends, client/provider, or competitors (the process would certainly end with different results).

The main common point between HA and our assistants is the behaviour extraction issue. HA reacts to silence detection and when the communication channel is idle for a too long time, HA contacts both users to figure out a new topic of discussion, if desired. Our assistants listen to all messages transferred to the system to extract as much information as possible, then to revise their beliefs and possibly react.

Another project is from Helmy et al. [20] about their Kodama agents assisting web search. Kodamas form an assistant team, specialised in retrieving information on Internet. Their collaboration and underlying interactions allow certain information retrieval performance and a flexible load distribution. The main difference with our project is most Kodamas serve different users simultaneously as local infrastructure. Also, Kodamas do not provide explicit explanation facilities so that users have no insight of the system mechanisms. They mainly exploit the user browsing history and the static profile to refine their search.

5.2 Negotiation Support Systems

Much recent work relies on agents that negotiate on behalf of human users. The typical protocol collects information from the user, performs the negotiation, confirms results with the user, and eventually commits the transaction [5]. These systems often improve time- and cost-efficiency in any kind of negotiation [1, 5], so that successful simple industrial versions were derived as EBay^{TM} [21]. The reason of this simplification is that most users lack faith in most types of automation that reduce their feeling of control. Consequently, users are often reluctant to the idea of letting artificial agents negotiate for real personal affairs and prefer self-performance with possible assistance as our proposal. In the commerce-centric NSS domain, Kasbah, AuctionBot and MAGMA are three reference architectures. Other solutions feature either similar functions or less expressive properties, so we focused only on these three ones. In addition, we discuss the particular case of ASPIRE as it has a similar stance as ours in NSS.

To begin, Kasbah is the first attempt to design an agent-mediated negotiation platform [6]. Buying and selling users send their negotiation agents with constraints. Agents negotiate by following a decay function (defining the 'strategy' of the agent depending on the function profile). This first platform was innovative but suffers from its early model. Agents just follow a function, and although they can autonomously complete negotiations, their lack of reactivity limits the range of application and we cannot really trust them. The work of Klein at al in [8] shows the drawbacks of that type of framework in terms of robustness and reliability.

AuctionBot is a client-server architecture that was used to organise online auctions between anyone endowed with a web-browser [3]. In this platform, buyers and sellers participate to auctions either with simple agents (idea of a proxy) that just inform owners about the state of their auctions (no autonomy), or exploit their self-developed agents based on the AuctionBot API to negotiate on their behalf. Users can join or create auctions, and choose the type among the most frequents. Although very flexible and comprehensive auction performer, this platform suffers from the same drawbacks as Kasbah. However, the strict auction protocols often allow enforcing the rules. As stated in [22], one important feature of the AuctionBot is its neutral stance toward sellers and buyers. This third-party ensures strict compliance to the auction protocol and so breeds higher trust in that kind of automated negotiation.

MAGMA represents one attempt to implement the marketplace metaphor in computing environments for the future electronic commerce world [5]. The framework aims at comprehensive models of actors and entities of marketplaces and so relies on multi-agent systems. Built on the experience acquired from the previous Kasbah and AuctionBot platforms, MAGMA embodies agents that negotiate for users and a complete environment including banks, balanced communication infrastructure (no central 'hub'), advertising, and software representation of physical goods to ensure coherence with real products and prevent from defrauding. The negotiation process exploited for this experiment is only the Vickrey auction, described in detail in their paper. The realistic virtual market goal lets perform direct negotiations, but the main focus is on agent-brokered processes that act similarly to the AuctionBot. The comprehensive set of features implemented by MAGMA clarifies the concept of electronic automated negotiations. The experimental restriction actually ensures higher degree of trust in the agent reactions owing to the strict auction rules, and perhaps because the Vickrey protocol best practice to maximise one's utility is 'to be honest'. This project shows trust can be achieved by clear rule enforcement in a dedicated electronic institution. However, such infrastructures is still in the long term and the participative assistance can be an alternative meanwhile.

Finally, ASPIRE turns delegated negotiation into collaborative negotiation [7]. This collaboration seems matching better the idea of assistant that provides support or aid (inspired from Merriam-Webster online dictionary) rather than the 'proxy' represented by the delegated negotiation agent. Thus, ASPIRE provides an asynchronous negotiation platform enabling two parties negotiating any goods by exchanging messages and negotiation field values (price, etc.). Personal assistants intervene for instance when a user submits a message. Before letting the system relaying the message to the opponent, the agent detects some inconsistencies or unusual negotiation behaviours such as an over- or under-evaluated offer compared to the previous one. The agent features other functions that allow correcting potential strategic errors. ASPIRE is in the direction of our work and shows promising results along practical experiments conducted in simulations. However, we think the engineering of ASPIRE may be a limiting factor. The project intends to extend the personal assistant abilities to larger 'communication channels' (multimodal) with the owner but also other agents. Our multi-agent framework encompasses inherently these functions with assistant teams that intrinsically interact with both human, team members, and soon other agents. Furthermore, it covers more issues including user service, explanation facilities and other engineering concerns.

6 Conclusion

In this paper, we presented a team of personal agents that supports one user in electronic negotiations. The agent team replies to user request and also takes initiatives as would do efficient assistants. The latter functions rely on the principle of listening to user interactions with the outer world, by means of the recent concept of overhearing. Teams offer the advantage to diversify functions of agents, exploit the intelligence emerging from their interactions, and engineer incrementally the agent population, i.e. extend and modify available services. Through an example, we laid out our current status in describing the key mechanisms in the interaction protocols of these MAS.

As stated along this paper, the present model of our assistant teams for negotiation support is an original platform compared with both NSS systems and with assistance solutions. We identified the central endeavours as the study of interactions, knowledge representation and extraction from these interactions and the consequent reasoning issues.

Our ongoing work first addresses the development of a generic assistant interaction protocol verifying the properties introduced in section 4. Second, the example presented in this paper featured three agents. We are investigating which services are pertinent in the negotiation process to answer user needs. Thus, we intend to integrate for example a history agent as negotiation requires knowledge of the past to both avoid reproducing errors and adapt to new situations [23]. Also, the abstraction layer we proposed in this paper to let agents understand the semantics of natural language messages from users suffers from its interfering process. We aim at a more transparent mechanism that would interrupt the user only on critical dilemma and to base parsing on pre-defined negotiation ontology. Finally, agents cooperate with each other and the user in our present model. Next, they will be authorised to compete with other systems on issues the owner considers secondary and releasable to agents, thus bridging participative and delegated negotiation techniques.

Acknowledgement

We would like to thank Nicolas Sabouret for our fruitful discussions on this research, the anonymous members of the PRIMA'04 program commitee that reviewed this paper, and the editors and authors of this volume for their advice.

References

1. Chalupsky, H.; Gil, Y.; Knoblock, C.A.; Lerman, K.; Oh, J.; Pynadath, D.V.; Russ, T.A.; Tambe, M.: 'Electric Elves: Applying Agent Technologies to Support Human Organisations', 13th IAAI Conference, pp.51-58, 2001.
2. Rich, C.; Sidner, C.: 'COLLAGEN: When Agents Collaborate With People', First International Conference on Autonomous Agents, 1997.
3. Wurman, P.R.; Wellman, M.P.; Walsh, W.E.: 'The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents', Second International Conference on Autonomous Agents, 1998.
4. Alty, J. L.; Griffiths, D.; Jennings, N. R.; Mamdani, E. H.; Struthers, A. and Wiegand, M. E.: 'ADEPT - Advanced Decision Environment for Process Tasks: Overview and Architecture', BCS Expert Systems 94 Conference, pp.359-371, 1994.
5. Tsvetovatyy, M.; Gini, M.; Mobasher, B.; Wieckowski, Z.: 'MAGMA: An Agent-Based Virtual Market for Electronic Commerce', Journal of Applied Artificial Intelligence, Vol. 11, $N^o 6$, pp.501-523, 1997.
6. Chavez, A.; Maes, P.: 'Kasbah: An Agent Marketplace for Buying and Selling Goods', First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 1996.
7. Kersten, G.; Koszegi, S.T.; Vetschera, R.: 'The Effects of Culture in Anonymous Negotiations: Experiments in Four Countries', 35th HICSS, 2002.

8. Klein, M.; Faratin, P.; Sayama, H.; Bar-Yam, Y.: 'Protocols for Negotiating Complex Contracts', IEEE Intelligent Systems, pp.32-38, Nov./Dec. 2003.
9. Drogoul, A.; Vanbergue, D.; Meurisse, T.: 'Multi-Agent Based Simulation: Where are the Agents?', MABS02 Workshop, AAMAS'02 Conference, 2002.
10. Busetta, P., Donà, A., Nori, M.:'Channeled Multicast for Group Communications', First International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'02 Conference, 2002.
11. Gutnik, G. and Kaminka, G.: 'Towards a Formal Approach to Overhearing: Algorithms for Conversation Identification', AAMAS'04 Conference, 2004.
12. World-Wide Web Consortium: The Semantic Web: http://www.w3.org/2001/sw/
13. FIPA Specifications: FIPA Modelling: Interaction Diagrams, Working Draft, 2002.
14. FIPA Specifications: FIPA Communicative Act Library Specification, 2002.
15. Parsons, S. and Jennings, N.R.: 'Negotiation through argumentation - a preliminary report', Second International Conference on Multi-Agent Systems, 1996.
16. Ramchurn, S.D.; Jennings, N.R.; Sierra, C.: 'Persuasive Negotiation for Autonomous Agents: A Rhetorical Approach', IJCAI Workshop on Computational Models of Natural Argument, 2003.
17. Sabouret, N.: 'Representing, requesting and reasoning about actions for active components in human-computer interaction', LIMSI-CNRS, Report 2002-09, 2002.
18. Vargas-Vera, M.; Motta, E.; Domingue, J.; Buckingham Shum, S; Lanzoni, M.: 'Knowledge Extraction by using Ontology-based Annotation Tool', ACM, 2000.
19. Isbister, K.; Nakanishi, H.; Ishida, T.; Nass, C.: 'Helper Agent: Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space', CHI'00, 2000.
20. Helmy, T.: Amamiya, S.; Amamiya, M.: 'Collaborative Kodama Agents with Automated Learning and Adapting for Personalised Web Searching', 13th IAAI Conference, pp.65-72, 2001.
21. Ebay™ Web site: http://www.ebay.com
22. Han, B.; Lim, J.: 'Influence of Culture and Explanation Facility on Performance of Negotiation Agents', 35th HICSS, 2002.
23. Tzu, S.: 'The Art of War', Oxford Press, June 1971.