Development of a Multi-Agent System for cooperative work with network negotiation capabilities

Francisco Garijo[1], Juan Tous[1], José M. Matias[1] Stephen Corley[2], Marius Tesselaar[3]

[1] Telefónica I+D, Emilio Vargas, 6, E-28043 Madrid, Spain.
e-mail: fgarijo@tid.es

[2] BT Laboratories, Martlesham Heath, Ipswich, UK. IP5 3RE.
scorley@everest.srd.bt.co.uk

[3] KPN Research, P. O. Box 15000 ,9700 CD Groningen, The Netherlands.
M.J.A.Tesselaar@research.kpn.comv

Abstract. This paper describes the architecture and operation of a Multi Agent system for providing end users with an intelligent interface for video conference and cooperative work services. The system consists of negotiating Agents: the Personal Communication Agents offers the user an intelligent interface to the service, it negotiates the best conditions in terms of quality of service and costs, the Service Provider Agent (SPA) supports the provisioning of telecommunication services to customers; the Network Provisioning Agent (NPA) provides network connectivity upon requests from the SPA. The physical setting is made up network nodes interconnected through two Public Networks; the Internet and ISDN. Network nodes are based on PCs running Windows 95 /windows NT, and Unix work stations. The system is being developed as part of the EURESCOM P712 project. The objective of the project is to assess and make recommendations on the applicability of intelligent and mobile agent technology to telecommunications service and network management. The evaluation criteria and the current status of the system are presented in the paper.

1 Introduction

Recent progress in network infrastructure and distributed processing has made possible the marketing of new telematic services incorporating "intelligent features" according to user demands. Agent oriented technology might be an important source of technical solutions for modeling and implementing those services. Existing agent based services on the Web are useful examples showing the applicability of agent technology for the design and implementation of commercial services. Nevertheless, combining agent technology - concepts, techniques, methods and tools- with existing service and software technology gives rise to a number of open issues that need careful evaluation. The meaning of terms like 'agent', 'intelligent agent', 'agent based architectures', 'agent development', etc., have been widely discussed [1] [5] [13][21]. Other engineering issues such as, agent specification, agent communication, agent engineering, agent development tools, etc., have also been presented [10][13][20]. Unfortunately the advantages of agent based proposals, over existing protocol specification and implementation, and other distributed processing solutions are not clear. From an industrial point of view the potential, of agent oriented technology needs to be demonstrated and evaluated in practice with prototype applications, in order to asses its maturity and the possible risks. This is the main objective of the EURESCOM (European Institute for Research and Strategic Studies in Telecommunications) Project P712 "Intelligent and mobile Agents and their applicability to service and network management". The evaluation approach taken by the project consists on building prototype systems using agent based solutions. Evaluation data will be gathered by carrying out a number of experiments directly on the prototypes. In addition, equally valuable data will be provided by the experience of building the prototypes. Two cases study have been identified in the areas of maintenance and dynamic connection management. Each case study defines the functionality to be implemented, the scenarios to be demonstrated and the experiments to be carried out. This paper describes the configuration case study covering dynamic connection management. The Multi Agent System provides multimedia meeting services to mobile users traveling around the world. The connections needed for service provision are negotiated and allocated dynamically according to users needs. The experimental Multi Agent setting, and its functionality is described in section 2. including agent interaction and the agent environment. Agent's design is presented in section 3. It is based on distributed object oriented principles, incorporating the session concept for peer to peer agent communication. The evaluation parameters are described in section 4, and section 5 draws the conclusions and the current status of the project.

2 Configuration Case Study

The experimental Multi Agent setting is based upon the scenario described in FIPA Part 7 [9] which focuses on the agents negotiation capabilities for dynamic connection management. The Multi Agents' System provides multimedia meeting services to end customers traveling to different locations and wanting to contact other colleagues in different cities around the world. The user connects his/her portable PC to the hotel's local telecommunication resources, and activates the Personal Communications Agent (PCA). When the PCA is activated, it first registers its location to the Local Registration Authority Agent (LRA). The user could then ask the PCA to establish the multimedia meeting within 5 minutes, for the duration of approximately 2 hours, and within a budget of $500."

The agent asks the LRA for the addresses of Service Provider Agents (SPAs), and then starts to contact them to see what possibilities are available. The SPAs will then contact different Network Provider Agents (NPAs) to see what offers are available to set-up network connections. Once the SPAs have found suitable NPAs that can provide the service, they will make provisional bookings and report back to the PCA. The PCA will then select the most suitable SPA to provide the service. The contract is now 'signed' between PCA and SPA, and the network connection is activated by the SPA. The SPA will then convert the provisional booking into 'contracts' with the different NPAs.

During the lifetime of the service, the PCA will be actively monitoring the fulfilment of the contract and log any deviations.

The agents involved in the service and the network setting are represented in figure 1

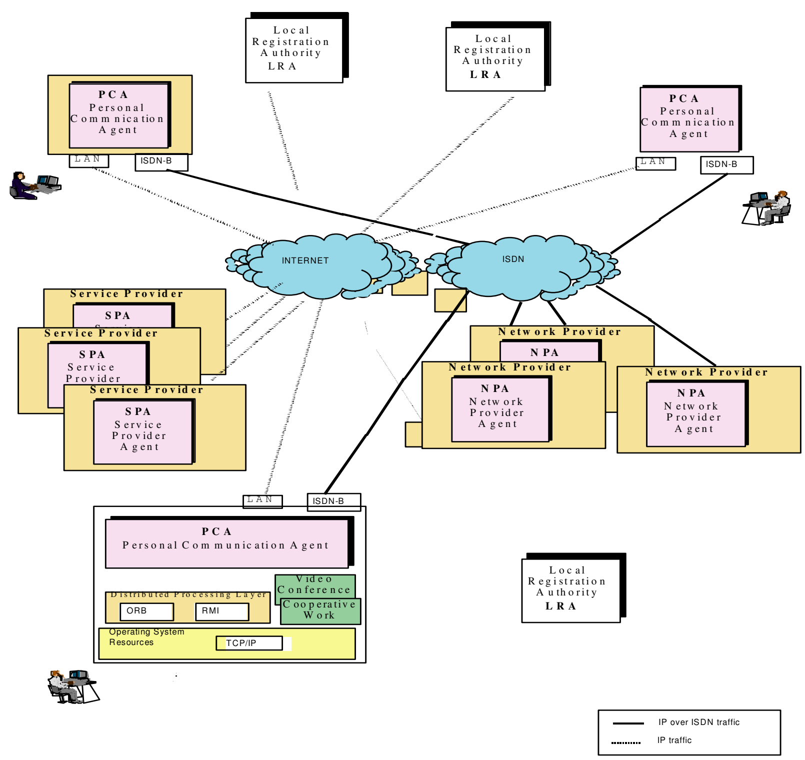

Fig. 1. Multiagent Setting

2.1 Agent Environment

The physical setting is made up network nodes interconnected through two Public Networks; the Internet and ISDN. Network nodes are based on PCs running Windows 95 /windows NT, and Unix work stations. The PCA communicates with other agents using the facilities of the Distributed Processing Environment (DPE) (ORB, RMI, etc.). The DPE protocols are based on TCP/IP. TCP/IP over Internet connections will be used for the agent' dialogues, while ISDN will be used to set up the connections needed to support video conference and cooperative work.

Collaboration with EURESCOM P715 has been established to enable the use of the ISDN European Services Platform (ESP) that is available between the project partners. The ESP is used to demonstrate provisioning of ISDN services using CORBA.

The software agent platform is based on Java and the Voyager platform version 2.0. [21]. PCA Inference capabilities are based on Ilog Rules Java [12]. The agents will migrate between the different partners over the Internet using the Voyager mobility services. This is particularly useful for the SPA that will travel between different locations to ease the negotiation process.

2.2 Agent Functionality

2.2.1 Personal Communications Agent (PCA)

The PCA represents the customer in it’s dealings with Service Providers. The key functions performed by the PCA during service provisioning are as follows :

- Elicit and validate local telecommunication service resources such as-connection points, terminals, etc. -.
- Elicit and validate service participants' destination and service constraints for example, - starting time, duration, cost, etc.-
- Negotiate with Service Providers Agents (SPAs) the best cost-benefit balance for service provisioning in terms of the constraints and preferences defined by the user.
- Provide the user with monitoring capabilities to supervise the negotiation process, or just wait for a message to initiate the service.
- Enhance user profile through dialogue input. The PCA should use the user profile to minimize the dialogue with the user, asking questions like “would you contact the same people as you did the last time you where here?”.
- Explanation capability. Inform the user why the service cannot be established. If some of the partners cannot be reached, the PCA should also inform the user , asking whether the service should continue or stop.
- Exhibit learning capability based on the classification of previous cases. Classification criteria are based on users preferences, list of partners, user locations, etc.

The PCA works autonomously on behalf of the user for service negotiation and service provisioning. User oriented functionality and negotiation facilities involves collaborative dialogues; PCA to User, and PCA to SPA.

2.2.2 PCA Interactions

User-PCA interaction. The dialogue between the user and the PCA is performed through the computer screen using visualization windows based on text, menus icons, active buttons, etc. The PCA should provide a robust user-friendly interface facilitating:

- Agent utilization: Configuration, activation, end, start, stop, monitoring.
- Acquisition and validation of service data: Service participants, service scheduling service cost, etc.
- Service control: Service activation, service interruption, cancel, re-start, etc.
- Service monitoring.

PCA-SPA Interaction. The PCA aims to negotiate with SPAs the best price/quality ratio for the service required by the user. PCA-SPA negotiation conversations are made up of messages sequences to:

- Communicate service requests.
- Get service offers.
- Communicate the selected SPA.
- Accept a service contract.
- Accept counter-offers.
- Execute contract.
- Resume contract.

PCA-LRA Interaction. The LRA manages Agent localization data. The PCA interacts with the LRA to:

- Register when it changes its location.
- Get the list of service providers.

2.2.3 Service Provider Agent (SPA).

Each Service Provider Agent represents the interests of a telecommunications service provider and supports the provisioning of telecommunication services to customers. It adopts two distinct roles :

-Client of network services offered by NPA

-Provider of a variety of telecommunication services to end customers.

The key functions performed by the SPA during service provisioning are:

- Authenticate the user.
- Determine component software/network service requirements.
- Negotiate with the PCA, terms and conditions of the delivery of the service.
- Identify secure Network Provider Agents (NPAs) for component services.
- Negotiate with NPAs for component network services specifying quality of service, bandwidth, source, sink(s), etc. The SPA tries to find the optimal solution in terms of quality of service and cost for providing the service to the end-user (through the PCA).

The SPA communicates with the PCA and with the NPA.

2.2.4 Service Provider Agent interactions

SPA-PCA Interaction. The SPA communicates with PCAs to:

- Negotiate service contracts.
- Start contracted services.
- Invite parties involved in contracted services to participate.
- Provides the configuration services upon requests from PCAs.

SPA-NPA Interaction. The SPA uses the NPA to provide the PCA with the network connections needed for the requested service. The interaction consists of:

- Receiving requests from a PCA
- Translating and relaying the requests into call-for-proposals to one or more NPAs
- Collecting the responses from NPAs
- Negotiating contracts with NPAs
- Requesting the network connectivity
- Activating services
- Terminating the network connectivity after the termination of the configuration service.

2.2.5 Network Provider Agent (NPA).

The Network Provisioning Agent (NPA) represents a network domain. Its major responsibility is the provisioning of network connectivity upon requests from the SPA.. For this purpose, the NPA has to interact with the SPA representing the customer, the network management system representing the local network domain and with other NPAs representing other network domains in the global environment.

The key functions performed by the SPA during service provisioning are as follows:

- Negotiate with the PCA, terms and conditions of the delivery of the service.
- Negotiate with other NPAs, terms and conditions of external connection segments.

To provide the requested connection, the NPA will have to first break down the task into local and external connection segments, based on some service strategy and knowledge about the global network environment. Then it will try to reserve connection segments in its local domain and request other NPAs to reserve segments in their domains in order to connect the sources and destinations

2.2.6 Network Provider Agent interactions

NPA-SPA Interaction

The NPA offers the network connection service to the SPAs via the following capabilities:

Processing the call-for-proposals from the SPAs.

Generating bids to the SPAs.

Providing network connectivity upon setup request from SPA.

Updating the network connectivity following the requests from SPA.

Releasing the network connectivity after the termination of the configuration service.

NPA-NPA Interaction

Other NPAs offer network connection service to the requesting NPA.

At this moment, we propose to adopt the same interaction protocol as NPA-SPA and SPA-NPA for NPA to NPA' interaction. This will be refined in the future after identifying the specific interaction.

2.2.7 Local Registration Authority (LRA)

The LRA plays the role of service broker. It provides service providers with information and users' location data. Interactions between agents and LRA follow the client-server model. The key functions performed by the LRA during service provisioning are :

- SPA registration.
- PCA registration when it changes its location.
- Provision of the list of service providers.
- Provision of location information of service participants' PCAs.

3 Prototype Design

The configuration prototype design is defined using an OO approach see figure 2-.

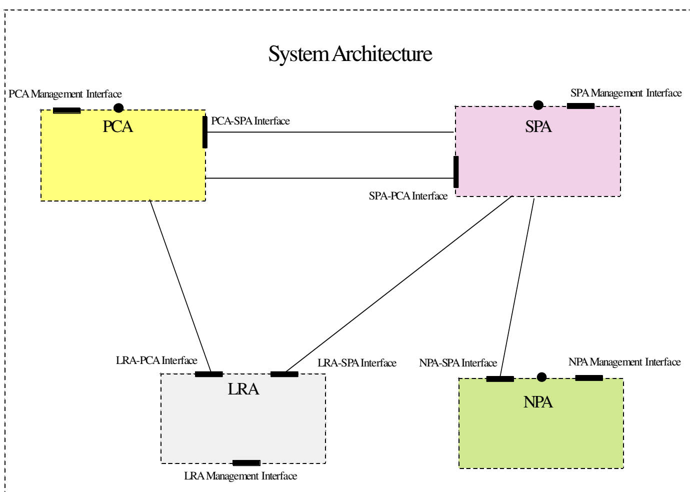

Fig. 2. System Design architecture

At the highest level the prototype is defined as a system formed by several subsystems: PCA, SPA, NPA, and LRA. There will be more than one instance of PCA, SPA and NPA. Each agent provides two interface types. The management interface contains the opertations needed for agent management such as agent monitoring, agent configuration and agent control. Agent to agent interfaces contain the operations needed to perform peer to peer communication.

Table 1 provides a summary of the operations included in each agent-agent interface.

Interface Operation	Client	Description
Interface PCA		
AceptOffer(bid_ref#,offer)	SPA	SPA send offers for a bid proposal
AceptContract(offer_ref#)	SPA	SPA send contract for a service
ServiceInvitationRequest(ref#, service type, quality parameters, calling party)	SPA	SPA invites a PCA to join in a service requested by another PCA
Begin_Service(contract#, participants' IP number list, service parameters)	SPA	activates the selected service contract and configures user's hardware
Interface SPA		
Bid_Request(ref#, service type, quality parameters, offered price, bid validity time, user list)	PCA	Ask a SPA for an offer for a specified service
Contract_Request(offer_ref#)	PCA	Ask a SPA for a contract for a specified service
Activate_Service_Request (contract#)	PCA	Requests the SPA to start a service
ConfirmService(contract#)	PCA	Ask a SPA for a service confirmation
CancelService	PCA	Ask a SPA to cancel a service
End_service(contract #)	PCA	Ask a SPA to end a service
Interface NPA		
ConfigureService(CID, QoS)	PCA	ask NPA if a connection necessary to deliver a service to PCA can be fulfilled
TearDownService(CID, QoS)	PCA	Teardown the connection
ConfigureConnection(QoS, CID)	PCA	Activate connection with QoS
ConfirmConnection(CID)	PCA	NPA can provide desired connection.
Interface LRA		
Register(agent_id)	PCA	PCA registers itself with a LRA
Deregister(agent_id)	PCA	PCA deregisters itself with the LRA
GetSPAList()	PCA	Get a list of known or available SPAs
GetUserLocation(user)	***SPA,PCA***	Get the location of the user

3.1 PCA Architecture and components

the TINA session concept [20] to achieve parallel interactions among different subsystems. It consists of the following subsystems.

Agent Management. Its goal is to achieve the functionality associated with the management interface. It will not be implemented in the case study

Task Manager. The Task Manager controls the PCA behaviour. It processes the inputs received through the Management interface and the PCA-SPA interface, tracking the global agent state, and monitoring the behaviour of the PCA. It also creates, monitors, and co-ordinates the sessions needed for

- User-PCA dialogue (User Session Manager).
- PCA-SPA dialogue (Service Session).
- Enhancing the user profile according to the service user (User Profile Manager).

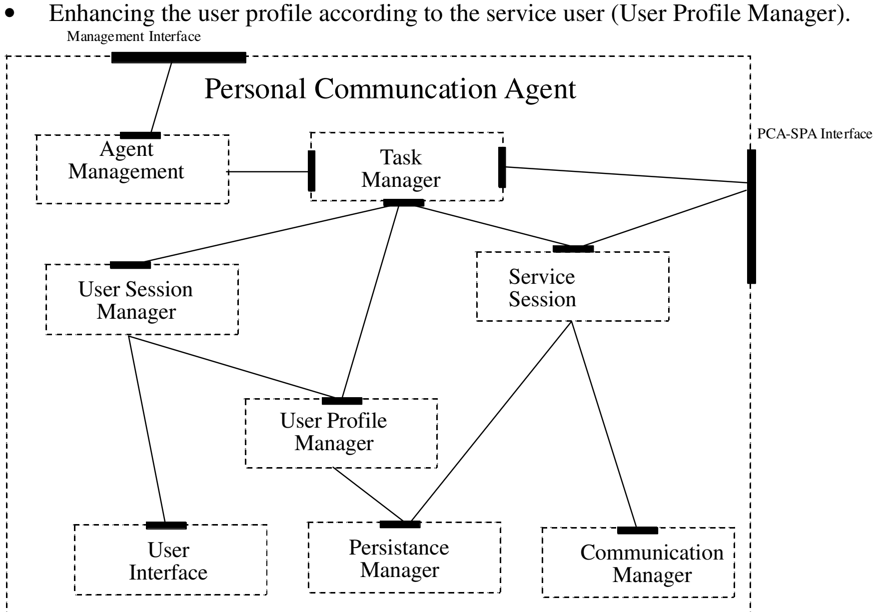

Fig. 3. PCA Architecture

User Session Manager. The User Session Manager processes the user inputs through the user interface. It manages and controls the dialogue with the user.

Service Session. The Service Session holds the computational resources needed to obtain the service functionality required by the user. The Service Session, controls the negotiation process with the SPAs, and set up the communication resources needed for the provision of the service. The Service Session uses the Persistence Manager to stored its internal state.

User Profile Manager. The User Profile Manager (UPM) receives inputs from the Service Session to obtain user profiles. When the user sessions terminate, the UPM updates the user profile according to the service results.

User Interface. Represents the computational and graphical resources needed for user interaction

Persistence Manager. Provides a persistence services to the PCA

Communication Manager. Provides an interface to use and control the communication resources needed for service provision.

The **Task Manager** internal structure is in figure.4. The **Task Control Manager** (TCM), is represented by a symbolic processing module made up of Task Control Knowledge defined by means of plan libraries and an inference engine.

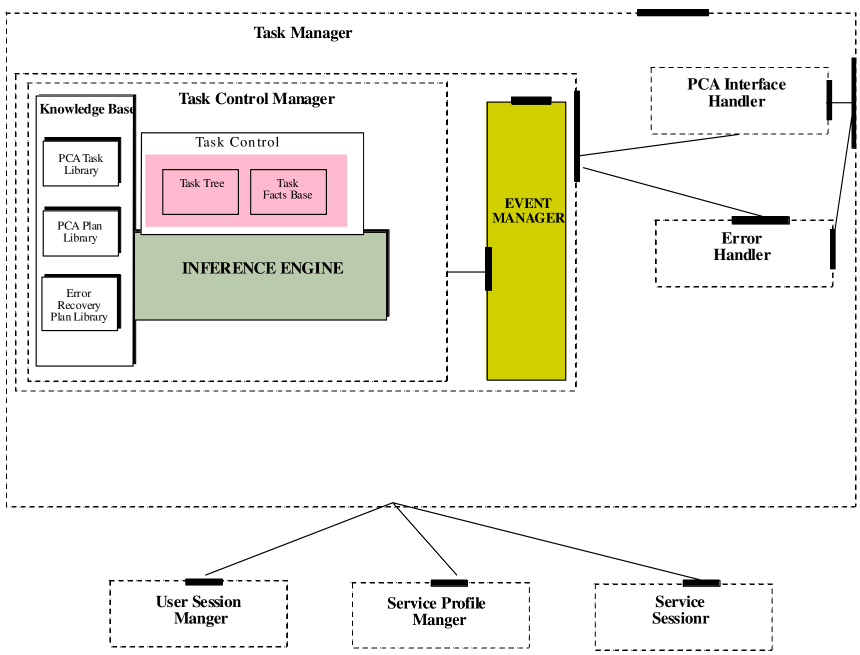

Fig. 4. Task Manager Architecture

The TCM receives events through the Event Manager. The reasoning engine uses the events to select plans from the plan library, creating a goal tree, and selecting an appropriate task to solve the pending goals. Task execution allows the control of the external resources asynchronously.

The User Session Manager (USM) creates and controls user sessions via the User Interface -Figure 5- The symbolic processing module contains the knowledge needed to guide and control the user-PCA interaction. USM KB is made up of service objects, information acquisition plans, input validation plans, and task resolution methods. The working memory is structured into two goal spaces: Service Monitoring and Control workspace keep track of the user input related to service monitoring , and Service definition control deals with Service definition and validation.. The USM uses the User Profile Manager to get the user profile when new service sessions start, and storing enhanced version of the user profile when service sessions terminate.

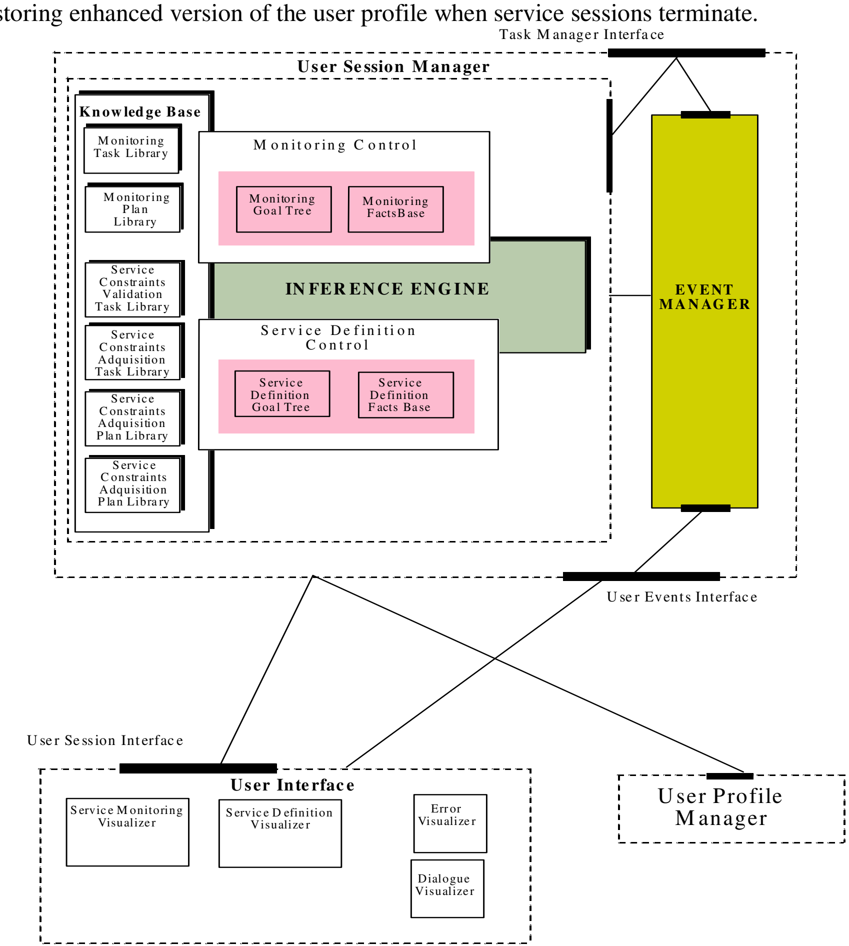

Fig. 5. User Session Manager Architecture

The structure of the Service Session is in shown in figure 6. The Service Session Control module creates, activates and monitors the Negotiation Session (NS), the User Service Participation Session (USPS) and the Service Control Session Resource

(SCSR). The NS performs the negotiation process with SPAs. The USPS achieves service control, and the SRCS provides the interfaces needed for resource management

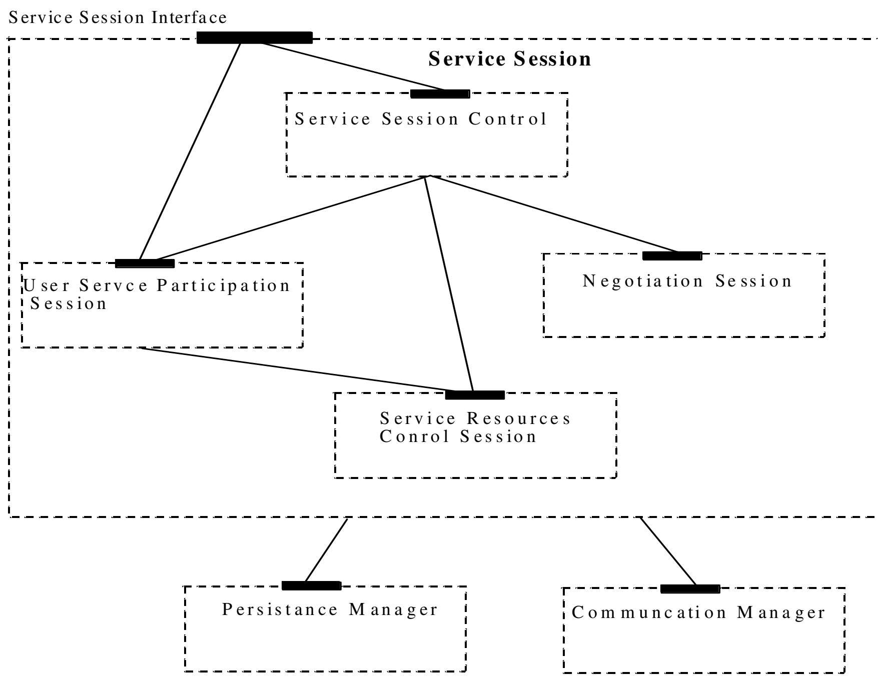

Fig. 6. Service Session Architecture

User Profile Manager. Its internal structure is based on a symbolic processor with specific knowledge to classify the data obtained from the SS. This new information is included in a new enhanced version of the user profile. The SS will use the enhanced version to improve the dialogue with the user in future service demand to the PCA.

4 Experimentation and evaluation

The metrics and the experiments to be carried out with the prototype are described in the following table. The experiments describe those aspects of agent operation that are to be evaluated and the metrics that need to be acquired in order to understand the use of agent technology.

The information obtained will be used to assess the applicability of agent' technology and to draw recommendations for its utilization in network and service management.

Metric	Experiments
Mobility aspects	Mobility aspects will be measured for the Personal Communications Agent and the Service Provider Agents. Agent platforms will have a time-stamping mechanism in the AMS that will leave time stamps when registering, deregistering etc. The information is sent in messages to the home agent platform where a test agent will further process the timing information.
Communication capabilities	Communication capabilities will be measured / characterized as applicable between the following agents: PCA – SPA and SPA - NPA Qualitative metrics are obtained by observation of the agent. Quantitative metrics are obtained using a measuring and counting mechanism that will: • Measure communication time in seconds as the end-to-end time for each message to be transmitted and processed by agents. • Number of messages per communication phase. • Complexity of problem as the number of steps required to solve the communication problem.
Life cycle rela capabilities	Creation time, will be measured for: The PCA that travels to agent platforms of service provider agents. Destruction time, will be measured for: The PCA after the connection has been terminated.
Security	Security issues will be identified for all agents which need to register with a specific agent platform.
FIPA conformanc	The overall prototype will be assessed according to the FIPA conformance checklist (assuming one exists by the time we come to complete the experiments).
Business benefit	Business benefit comparisons will be made between our agent solutions and conventional solutions for the following activities like: • manual setup of connections between parties • manual maintenance of the connections • manual tear down of the connections

5 Current status and further work

The project started in March 1997 and will end in March 1999. After an initial domain analysis activity the case studies were agreed and defined. Prototyping started in late 1997 and will continue until end July 1998. Two engineering approaches have been addressed for prototype implementation. One approach follows FIPA recommendations for agent specification and implementation. The second one, representing Telefonica's view in the project, is based on Open Distributed Processing Recommendation and Object Oriented Modeling. The Multiagent Architecture is modeled as a collection of Distributed Systems offering several interface types.

The justification of this approach follows from several considerations. Building commercial services will require agent technology to be integrated with conventional software technology, data base technology, and distributed processing. As existing methodologies, computational environments and development tools, allow this integration, it is important to carry out experiments to assess the development of systems satisfying the requirements and the functionality required for Multi Agent Systems. Most of the new engineering concepts defined by OMG and TINA seems applicable to agent based systems. Unfortunately people developing conventional telecommunication services are not familiar with symbolic techniques. A pragmatic approach to introduce these techniques is to design the symbolic component as a conventional component, hiding the symbolic implementation, and avoiding side effects with the rest of the subsystems. This is a way to demonstrate in practice the need of cooperation between symbolic processing components, and imperative components, to provide enhanced functionality to end customers. On the other hand, new ODP concepts like the TINA session concept, are better suited to the specification and implementation of peer to peer conversations among computational entities, rather than conventional message based approaches. The TINA session facilitates dialogue specification, avoiding low level message handling and correlation during the implementation.

The evaluation experiments will provide the data needed to analyze the advantages and disadvantages of FIPA based approaches as well as ODP approaches. As prototype implementation is planned to be finish by the end of July, initial results about implementation issues and system operation might be reported and discussed in the workshop.

Acknowledgements

The authors would like to thank the EURESCOM organisation, particularly Juan Siles, and other members of the Project team for their support and help in producing this paper.

Disclaimer

This document is based on results achieved in a EURESCOM Project. It is not a document approved by EURESCOM, and may not reflect the technical position of all the EURESCOM Shareholders. The contents and the specifications given in this document may be subject to further changes without prior notification. Neither the Project participants nor EURESCOM warrant that the information contained in the

6 References

[1] J. Bradsahaw, "Software Agents", : AAAIPress/The MIT Press, 1997
[2] Davis and Smith, R.G (1983). Negotiation as a Metaphor for Distributed Problem Solving. Artificial Intelligence 20(1): 60-109
[3] B.Chaib-draa. Industrial Applications of Distributed A.I.. Communications of the ACM, Vol 38, N.11, pp:49-53, Nov. 1995
[4] Y. Demanzeau. From Interactions to Collective Behavior in Agent-Based Systems. 1st European Conference on Cognitive Science, Saint-Malo, France, avril 1995
[5] M.R.Genesereth, S.P.Ketchpel. Software Agents. Communications of the ACM. Vol 37, 7. pp:49-53 July 94
[6]EURESCOM P712 web page http://www.eurescom.de/public/projects/P700-series/P712.HTM
[7] S. Hedberg Agents for sale: firs wave of intelligent agents go commercial. IEEE Expert Dec 1996 pp 16-23
[8] FIPA Web Page, http://drogo.cselt.it/fipa
[9] FIPA. Application Design Test: Personal Assistant. FIPA´97 Draft Specification: Part 5. Revision 2.0.P. Kearney, D. Steiner, editors.1997
[10]Garijo F.J., Hoffman D. A Multi-Agent Blackboard Based Architecture for Operation and Maintennance of Telecommunication Networks. Proc of Avignon 1992 pp 427-438. Jun 1992
[11] Hylton, Jeremy, Ken Manheimer, Fred Drake, Roger Masse, Guido van Rossum, and Barry Warsaw "Knowbot Programming - System Support for Mobile Agents", Sept, 1996
[12] Ilog Rules http:// WWW.Ilog.com
[13] Maes, P., "Agents that reduce work and information overload", Communications of the ACM, 37(7): 31-41, July 1994
[14] Object Management Group, “http://www.omg.org/”.
[15] C. J. Petrie Agent Based Engineering, the Web, and Intelligence. IEEE Expert Dec 1996 pp 24-29
[16]Pruitt, Dean G (1981). Negotiation Behaviour. Academic Press.
[17] K. Sycara, A. Pannu, M. Williamson, D.Zeng, K Decker. Distributed Intelligent Agents. IEEE Expert Dec. 1996pp36-45
[18]Sycara, K.P (1989a). Argumentation: Planning Other Agents' Plans. Proc. of IJCAI, Detroit, pp 517 - 523.
[19] Velthuijsen, H. and Griffeth, N., "Negotiation in Telecommunications Systems", In Notes AAAI Workshop on Cooperation among Heterogeneous Intelligent Systems, pages 138-147, San Jose, CA, 1992
[20]Telecommunications Information Network Architecture, “http://www.tinac.com”.

[21] Tina7 C.Abarca, P.Farley, J.Forslow,J.Garcia. T. Hamada, P.F.Hansen, S.Hogg, H.Kamata, L.Kristiansen, C.A. Licciardi, H.Mulder, E.Utsunomiya, M.Yates. Service Architecture V 0.4 April 1997 TINA-C Deliverable

[22] Voyager Platform http:// WWW.objectspace.com/voyager

[23] K.Urzelai, F.J.Garijo. MAKILA: A Tool for the Development of Cooperative Societies. Lecture Notes in Artificial Intelligence 830 (Proc. of MAAMAW´92). pp: 311-323,1994.

[24] White, J. Mobile Agents White Paper. General Magic Inc. http://www.genmagic.com/agents/Whitepaper/whitepaper.html

[24] M. Wooldridge, N.R. Jennings. Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2), 115-152, 1995