Agentcities

Agent Technology Exhibition

ATC03

Barcelona 6 february 2003
Campus Nord
Universitat Politècnica de Catalunya

organised by:
the IST Agentcities.NET project

Exhibition & Competition

IST-2000-28385
European Commission 5th Framework

Agentcities.NET

German Research Centre for Artificial Intelligence
Motorola
BT Exact Technologies
Ecole Polytechnique Fédérale de Lausanne
Universitat Politècnica de Catalunya

IST-2000-28384
European Commission 5th Framework

Agentcities.RTD

Motorola
Adetti
AEGIS
Agentscape
BTExact Technologies
Broadcom Eirean
Communication Technologies
Ecole Polytechnique Fédérale de Lausanne
Fujitsu
German Research Center for Artificial Intelligence
Imperial College
Queen Mary College
Universitat Politècnica de Catalunya
University of Parma
Telecom Italia Lab

[Cover image: Chimneys on top of Antoni Gaudi's *La Pedrera*, emulating a troop of soldiers.]

Table of Contents

Introduction, Welcome and Thanks

Welcome to the Agent Technology Exhibition at iD3 – the event combines the finals of the Agentcities Agent Technology Competition (ATC'03) with a series of invited demonstrations and a lightweight program of talks. The demos represent a snapshot of the state of the art in the area of Agent technology, open systems and Internet technologies.

The motivation for the exhibition arose out of the wish to showcase for the entries submitted to the 2003 Agentcities Agent Technology Competition and to provide a forum for researchers and industry in the area to demonstrate their systems. The event will culminate with *the Competition Reception and Prize Ceremony* that will be held at the Barcelona City Hall. Invitations are open to everybody but you must bring your printed paper invitation for security reasons.

Demos are contributed by both Universities and Industry groups including a significant number from outside of Europe. Many of the European groups are partly funded through the Agentcities.NET deployment grant program or other European Commission research.

We would like to thank all the exhibitors taking part and adding such excellent content to the event. Particular tribute goes to all the teams that entered the technology competition – putting such a great deal of effort to produce the selection of demo finalists we have at the event. Organisation of the competition itself would not have been possible without the contributions of many people: the testers, the reviewers, the judges and most importantly the European Commission IST program itself which provided the funding for the prizes and manpower for organisation.

A special word of thanks is reserved for our many sponsors who have contributed with great enthusiasm to support the event – Fujitsu-Siemens, Motorola and the City of Barcelona have been particularly generous as *Main sponsors* by providing us with great hardware to augment the prizes. The student prize was created jointly by the Asociación Española para la Inteligencia Artificial (AEPIA), Fundació Catalana per la Recerca and iSOCO.

Welcome and enjoy!

David Bonnefoy & Steven Willmott

Competition

About the Agentcities Agent Technology Competition

Many types of projects have their place in the Agentcities initiatives – large and small projects, projects by companies, universities or individuals… That's where the idea of a competition comes from: Agentcities.NET, which aim is to promote the development of Agentcities, created it to attract novel ideas in a not so usual environment.

The competition was open to any participant worldwide. Two categories were created: infrastructure (support for the development of the network) and applications (value-added agent services). In addition, a Special Prize will be awarded to an entry deployed in the Agentcities network, and a Student Prize was created by the Asociación Española para la Inteligencia Artificial (AEPIA), Fundació Catalana per la Recerca and iSOCO.

The European Commission offered a generous budget for great prizes, which were even improved through sponsorship.

The competition was open in August 2002. The participants were asked to develop and document an agent system in the spirit of Agentcities – showing openness and innovation among others. Their entries had to be submitted, on paper, by November 15th. The entries were then assessed by a panel of reviewers, and in parallel tested by a panel of testers for those participating to the Special Prize.

The panel of Judges selected the finalists in each category at the end of the year, and these finalists were invited to come to Barcelona for the Technology Exhibition.

Always in the spirit of Agentcities, it was important that the systems were implemented and deployed. That is why the final is live, and the entries will now be presented to you at the Technology Exhibition.

The winners of the competition will be known soon – be sure to be present at the award ceremony Thursday night at the Barcelona City Hall!

The Judges

Joël Bacquet

Joël Bacquet is working at the European Commission in Brussels, in the business application unit of the Information Society technology (IST) programme. He is coordinating the “Business networking” area.

Bernard Burg

Bernard Burg is leading the Agent for Mobility Department at Hewlett-Packard Laboratories in Palo-Alto. He is one of the initiators of Agentcities. His domain of interest is the deployment of advanced web services.

Yves Demazeau

Yves Demazeau is currently CNRS Research Fellow and External Professor at the USD Faculty of Science and Engineering. He is the coordinator of the MAGMA research group on Multi-Agent Systems at Leibniz laboratory in Grenoble and external member of the MIP Institute in Odense.

Tim Finin

Tim Finin is a Professor in the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County (UMBC). He has over 30 years of experience in the applications of AI to problems in information systems, intelligent interfaces and robotics.

Nick Jennings

Nick Jennings is Professor of Computer Science in the Department of Electronics and Computer Science at Southampton University where he carries out research in agent-based computing. He is head of the Intelligence, Agents, Multimedia Group and is also the Chief Scientific Officer for Lost Wax.

Ramon López de Mántaras

Ramon López de Mántaras is Research Full Professor and Deputy Director of the Artificial Intelligence Institute of the Spanish Higher Council for Scientific Research. He is presently working on Case-Based Reasoning, on a Multi-agent approach to Autonomous Robots Qualitative Navigation, and on AI applications to Music.

Jörg Muller

Jörg Müller has been active in the area of agent technology for over a decade. He is currently leading an R&D group on agent technology at Siemens Corporate Technology in Munich.

Mark Randolph

Mark Randolph is Fellow of the Technical Staff and Director of the Mobile Platforms and Services Laboratory at Motorola Labs. He is responsible for research and development of future generation platforms and technologies that enable delivery of content and services to mobile devices.

The Reviewers

John Shepherdson
Mark D'Inverno
Eduardo Alonso
Frank Dignum
Robert Ghanea-Hercock
Jonathan Dale
Fabio Bellifemine
Paul Davidsson
Omer Rana
Monique Calisti
Paolo Petta
Jamie Lawrence
Antonio Moreno
Makram Bouzid
Erwin Bonsma
Chris van Aart
Heikki Helin
Dominic Greenwood
Hiroki Suguri
Ronnie Taib
Ruth Aylett
Denis Pierre
Valentina Tamma
Michael Luck

The Testers

(Special Prize)

Antonio Lopes
Jérôme Picault
Farooq Ahmad
Giovanni Rimassa
Emma Rollón
David Cabanillas
Josep M. Pujol

The Prizes

The Agentcities Agent Technology Competition has a series of excellent prizes funded by the EU Commission IST program (through the Agentcities.NET project grant) and generous contributions from competition sponsors - in particular the main prize sponsors:

- Fujitsu Siemens providing two of their B series ultra-portables and four Pocket LOOX PocketPC handheld devices for main and student prizes!
- Motorola providing their latest generation T720 Smartphones for the infrastructure and application prizes.
- Asociación Española para la Inteligencia Artificial, Fundació Catalana per la Recerca and iSOCO teaming up to create a new prize for Best Student Entry

(Please note that all prizes announced here are subject to change for items of equivalent value without notice at the discretion of the organisers. Hardware prizes cannot be exchanged for cash.)

Main Prize Categories

1st Prize

3000 Euros

One **B ultra portable sub-notebook** (Pentium III/800MHz, a 10.4 TOUCHSCREEN monitor, 256 MB RAM, an external CR-RW/DVD, 30 GB HD)
and three Motorola **T720 Mobile Phones** (with colour screen, polyphonic ringtones, GPRS, and Java programmable)

2nd Prize

2000 Euros

and two Motorola **T720 Mobile Phones**

3rd Prize

1000 Euros

Agentcities Special Prize

For those entries which are actually deployed in the Agentcities network there is an additional **SPECIAL PRIZE** for the best such entry. The special prize value is:

3000 Euros

The special prize may be won by any of the entries including winners in other categories (i.e. prizes are non-exclusive).

Student Prize

Sponsors AEPIA, FCR, iSOCO and Fujitsu Siemens have combined to offer an additional prize of:

1000 Euros

and two Fujitsu Siemens Pocket LOOX 600 PDAs

to the best entry submitted by teams where the majority of work was carried out by **students** and was lead by one or more students. (A student is defined as a person matriculated in a course of primary, secondary or higher education; leading the work means carrying out the majority of design, architectural and technical choices related to the project.)

The student prize may be won by any of the entries including winners in other categories (i.e. prizes are non-exclusive).

The Finalists

A1: CONSORTS

Cognitive Resource Managing Agents in Ubiquitous Computing

CONSORTS Team (Cyber Assist Research Center, AIST)
Koichi Kurumatani, Akio Sashima, Noriaki Izumi, Yutaka Matsuo, Shigeyoshi Hiratsuka
sashima@carc.aist.go.jp

Abstract

CONSORTS (Architecture for COgNitive reSOurce management with physically-gRounding agents) is a new kind of agent architecture for Ubiquitous computing, which is designed to realize mass-user support in addition to conventional personal assistance. The key concepts in Consorts are "physically-grounding" and "cognitive resources". By using sensory information brought by ubiquitous environment, agents have grounding to physical world and they are conscious of physical resources (especially spatio-temporal resources) in a cognitive way, i.e., they can recognize, reorganize, and operate raw physical resources as cognitive resources. Services realized in Consorts include 1) extension of conventional personal services using physical information such as position, and 2) mass-user support that realizes information providing and social coordination for mass-user as mass beyond personal support.

Challenge of Consorts Architecture

For the purpose of CARC (Cyber Assist Research Center, AIST) that pursuits human-centered information technology, we have been developing CONSORTS, an agent-based software architecture that mediates the real world and a digital world, for assisting human in accessing intelligent information services. The Ubiquitous computing emphasizes sensor-based physical information in the real world; the semantic web emphasizes ontological information resources in a digital world; Consorts architecture emphasizes harmony of the information between two sorts of the worlds.

In order to bridge the gap between the digital world and the real world, the Consorts agents roam both in the Ubiquitous computing environment and in the Semantic web based Web services. The Consorts agent provides the context-aware assists over the physical-location-based cognitive resources and the semantic-information-based digital services, in the cooperation with the orchestrated agents that manage the both sensor information of the physical environment and ontological information of the web services.

If participants of Agentcities provide our proposed services about location-based local information for Agentcities, an agent can access the real world information covering all over the world. It can also be a great demonstration of a capability of agents connected to open environments such as the Internet, Web services or Agentcities. From the academic standpoint of the agent research, our proposal can clarify a possibility whether Ubiquitous agents can be unified with open agent systems including Agentcities. Our challenge is aiming at achieving the paradigms of agents tomorrow: Agentcities, Ubiquitous computing, Web services and the Semantic web.

Outline of Consorts Architecture

The Consorts architecture consists of the following type of agents to realize the concept.

- **Spatio-Temporal Reasoner** manages "Spatio-temporal inference engine" and "Spatial Information Database". The database can be regarded as a kind of GIS specialized for the Ubiquitous-computing environment. Spatio-temporal reasoner realizes fine-grained context-awareness of agents by using high-resolution representation of Spatial Information Database. Spatio-temporal reasoner also provides basic service for mass-user support, such as navigation service for the mass-user.
- **Services Agents** provide the services, such as tour guide, navigation and information presentation. Services agents realize the flexible service by using semantic information on the Internet, especially Semantic Web and Web Services.
- **Personal Agent** manages to interaction to mediate between service agents and a user. The agent controls user interaction to provide information for the user. The personal agent orchestrates the various services of agents that manage sensor information of the physical environment and ontological information of the web services.
- **Device Wrapper Agents** hide the diversity of physical device and legacy application. Device Wrapper Agents realize device-independent applications by wrapping the raw information, which is derived from legacy devices and applications, with the standardized agent communication language.
- **User Model Manager** manages user model, such as user's attribute, preference, and intention.

Museum Scenario

Based on the Consorts architecture, we have implemented context-aware information assist service in a museum. In this scenario, the agents are aware of the distance between a user and a painting in the museum. The service scenario is that when the user approaches a painting, the agents automatically provide information of the painting via the user's portable display device. If the user needs more information of the picture, the user should push the "tell me" button of the portable device. The Consorts agents notice the user's request, make a search query about the picture with the necessary information they've already had, e.g. the picture's name and user's preference, and access the internet search engine, e.g. Google Web Services, instead of the user.

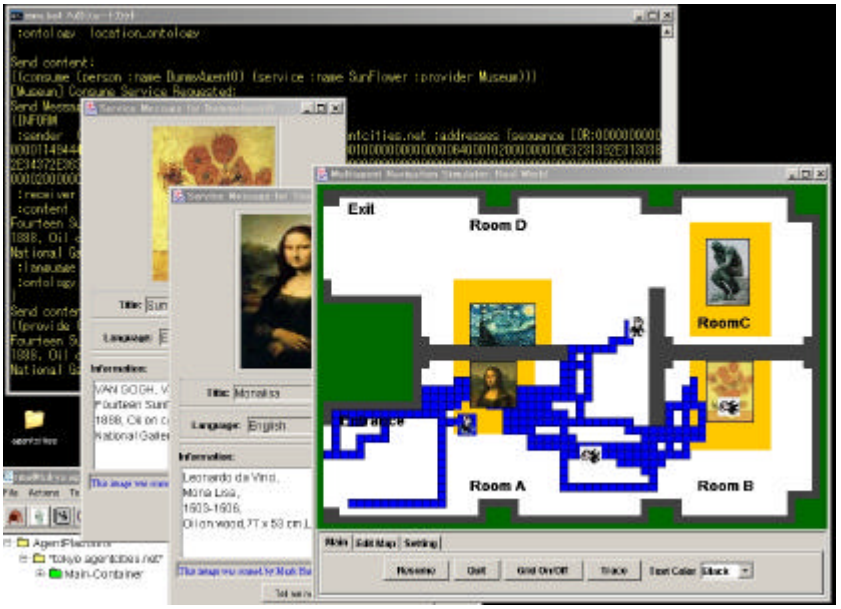


Fig. 1 Screenshot of the demonstration program

Fig. 1 is a screenshot of the demonstration program of museum scenario. You can see a museum map in a main window, two information windows with some pictures, and a message window displaying agent communication. In the map, human icons represent current locations of the users; the yellow zones represent service zones of the museum service agents; the blue lines represent users' trajectories. The information windows correspond to the screen of users' portable devices. In this demonstration program, the users randomly roam in the museum. If a user enters the yellow zone, an information window pops up in the screen to display the picture information.

You can run demonstration programs of Consorts as Java applets in our website (http://consorts.carc.jp/). If you need more information about the demonstration applets, please look at web pages in the website.

A2: GruSMA1

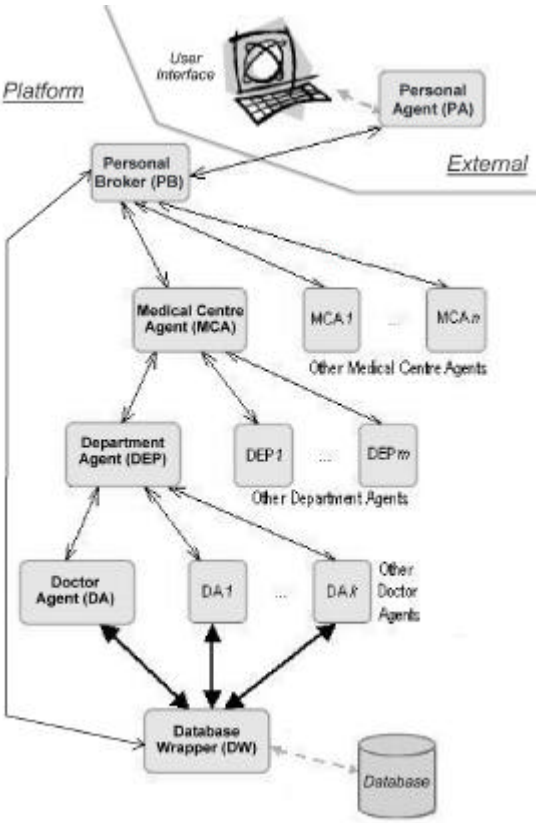
Deployment of agent-based health-care services

Computer Science and Mathematics Department,
University Rovira i Virgili
Antoni Moreno
amoreno@etse.urv.es

Abstract
In this project we have implemented and deployed an agent-based AgentCities platform that offers medical services to the citizens and visitors of a city. These services allow a user to a) make queries related to information about medical centres, medical departments and doctors, b) access his/her medical record and c) make an appointment to be examined by a particular doctor. At the examination time, the doctor is able to access and update the patient's medical record. Special attention has been paid to the definition of a medical ontology in RDF and to the implementation of mechanisms that guarantee the confidentiality of medical information.

Brief description of the system

We have implemented and deployed an AgentCities platform that contains agents that offer medical services to users (citizens and visitors of a city). The basic architecture of the system is shown in the following figure.



This multi-agent system contains six different types of agents: the user's *personal agent* (PA), the *personal broker* (PB), a *database wrapper* (DW) and some *medical centre agents* (MCAs), *department agents* (DEPs), and *doctor agents* (DAs). The information of a medical centre is divided in three levels. For each medical centre there is one MCA, several DEPs (one for each department of the centre) and several DAs (one for each doctor of each department). The MCA has the general information of the centre (e.g. its address and opening times).

Each DEP has the knowledge of a certain department (e.g. all the information of the ophthalmology department). Each DA maintains the schedule of a given doctor, and is aware of the doctor's visiting times. The personal agent provides a graphical interface of the MAS with the user. This interface is used to introduce the requirements of a search or to show the results of a query to the user. The personal broker is an agent that controls the access of the personal agents to the services provided by our system, by introducing authentication and avoiding direct communication with other agents. The user requests health care services to the personal broker. The database wrapper is the agent that controls the access to a database that contains the medical records of the users.

The system provides the following services:

- The user may ask for medical centres satisfying certain properties (e.g. those medical centres that are located in a particular city and have a cardiology department).
- The user is able to access his/her medical record.
- The user may make an appointment to be examined by a doctor in a specific medical centre and date.
- While an examination is being performed, the doctor accesses and updates the patient's medical record.

The system guarantees a secure access to medical records, so that only authenticated users and doctors that examine the users are allowed to obtain and update this information.

The main aim of the AgentCities initiative is to provide agent-based services that improve the quality of life of the citizens. The deployed application certainly contributes towards this objective, by giving to the users the possibility of accessing medical data in an easy and efficient way. The benefits of the system, were it to be implemented in a real setting, would reach both citizens and health care personnel.

Some examples of use of the system:

- A user that is located in a rural area, and needs to be visited by a local doctor. He may access his medical record by contacting the platform (e.g. just with a personal agent running in a PDA or a mobile phone).
- A doctor may use the system to access and update the medical record of a patient while he is performing a clinical examination of the patient.
- A visitor of a city may access the platform (e.g. from any PC connected to Internet) and find out information about the medical centres of the city.

IBundler: a decision support service for negotiating agents

"Los Chicos"
iSOCO, S. A.
Juan Rodriguez
jar@isoco.com

Abstract
Multi-item negotiations in industrial procurement pose serious challenges to buying agents when trying to determine the best set of selling agents' offers. Typically, a buying agent's decision involves a large variety of constraints that may involve attributes of separate items as well as attributes of different, multiple items. IBundler is intended as a negotiation service offered to buying agents to help them determine the optimal bundle of received offers based on their constraints and preferences. In this way, buying agents are relieved with the burden of solving too hard a problem (the problem can be proven to be NP[1]) and concentrate on strategic issues. IBundler is intended as a negotiation service for buying agents and as a winner determination service for combinatorial auctions. To the best of our knowledge this is the first agent service for such type of complex negotiation scenarios.

Extended description

Although the application of combinatorial auctions (CAS) to e-procurement scenarios (particularly reverse auctions) may be thought as straightforward, the fact is that there are multiple new elements that need to be taken into consideration. These are new requirements explained by the nature of the process itself. While in direct auctions, the items that are going to be sold are physically concrete (they do not allow configuration), in a sourcing event involving highly customisable goods, buying agents need to express relations and constraints between attributes of different items. On the other hand, a typical procedure is to buy a same product from different supplying agents, either for safety reasons or because offer aggregation is needed to cope with high volume demands. This introduces the need to express constraints on the number of supplying agents and the amount of business assigned to each of them. Not forgetting the provider side, supplying agents may also impose constraints or conditions over bids/offers. Offers are only valid if certain configurable attributes (quantity bought, delivery days) fall within some minimum/maximum values, and assembly or packing constraints need to be considered. Current CAS reviewed do not model these features. Without means of modelling these natural constraints, solutions obtained can be regarded as mathematically optimal, but unrealistic and with a lack of common sense, thus obscuring the power of decision support tools, and preventing the adoption of these technologies.

Next we detail the capabilities required by buyers in highly-constrained e-sourcing (negotiation) scenarios. The requirements below are intended to capture buying agents' constraints and preferences:

[1] Actually the problem faced by iBundler is more general than the winner determination problem in combinatorial auctions.

- **Negotiate over multiple items.** A sourcing event is usually started with the preparation of a request for proposal (RFQ) form, which describes in detail all requirements (e.g. volume, quality, delivery dates, drawings and technical documentation) for the list of items (goods or services) defined by the sourcing event.
- **Offer aggregation.** A specific item of the RFQ can be traded from several providers simultaneously, either because not a single provider can provide with the requested quantity at requested conditions or because buyer's explicit constraints (see below).
- **Business sharing constraints.** Buyers might be interesting to restrict the number of providers that will finally trade for a specific item of the RFQ, either for security or strategy reasons. It is also of usual practice also to define the minimum amount of business that a provider may gain per item.
- **Constraints over single items.** Every single item within an RFQ is described by a list of negotiable attributes. Since: a) there exists a degree of flexibility in specifying each of these attributes (i.e. several values are acceptable) and b) multiple offers referring this same item can be finally accepted; buyers need to impose constraints over attribute values. An example of this can be as follows: suppose that the deadline for the reception of item *A* is two weeks. This is a constraint individually imposed on each single unit of item *A.* However, although items may arrive any day within two weeks, once the first units arrive, the rest of units should arrive in no more than 2 days after. The latest is a constraint imposed to the total of product *A.*
- **Constraints over multiple items.** In daily industrial procurement, it is common that accepting certain configuration for one item affects the configuration of a different item, for example, when dealing with product compatibilities. Also, buyers need to express constraints and relationship between attributes of different items of the RFQ.

Analogously, next we detail the capabilities required by sellers. The requirements below are intended to capture selling agents' constraints and preferences.

- **Multiple bids over each item.** Sellers might be interested in offer alternate conditions/configurations for a same good, i.e., offering alternatives for a same request. A common situation is to offer price discounts per number of units. This means that a seller submits several offers and each offer only applies for a minimum (maximum) number of units.
- **Combinatorial offers.** Economy efficiency is enhanced if providers are allowed to offer (bid on) combination of goods. They might lower the price, or improve service assets if they achieve to get more business.
- **Capacity constraints.** Sellers need to specify that they will only participate in trading if a minimum (maximum) amount of business is assigned to him.
- **Packing constraints.** Packing units are also a constraint, in the sense that it is not possible to serve an arbitrary number of units (e.g. 100 units in 25-unit packages). On the other hand several goods are packed and served together, being not possible to buy a different number of units of each, and so holding a *homogeneity* constraint.
- **Complementary and exclusive offers.** Sellers usually submit *XOR* bids, i.e., exclusive offers that cannot be simultaneously accepted. Also, there may exist the need that an offer is selected only if another offer is also selected. We refer to this situation as an *AND* bid. For example, when pricing is expressed as a combination of basis price and volume-based price (e.g. 1000 units at €2.5 p.u. and then €2 each).

iBundler is offered as a decision support agent for highly constrained e-sourcing scenarios acting as a combinatorial negotiation solver for both multi-item negotiations and auctions. Thus, the service can be employed by both buying, negotiating agents and auctioneers in combinatorial auctions. *It extends current models by accommodating both operational constraints and attribute-value constraints. IBundler solver offers an XML language for expressing offers, constraints, and requirements. The XML specification is parsed into an MIP formulation and solved using available MIP solvers.*

Implementation / Use status

From its inception, IBundler was conceived to subsequently function as a software component, as an agent service, and as a web service. In this manner, the very same service can be exploited in different manners. On the one hand, the *IBundler solver* is currently integrated into *Quotes*, iSOCO's sourcing solution (http://www.isoco.com/en/content/solutions/solution_iquotes.html). *IBundler solver* allows Quotes to determine the optimal set of offers in combinatorial negotiations and auctions. As to negotiations, the result is offered to the buying enterprise conducting negotiations so that it can decide which providers to contract or continue negotiating with. As to auctions, IBundler solver helps the auctioneer solve the winner determination problem. IBundler is employed in *Quotes* to assist buying agents and solve combinatorial auctions.

IBundler has been also made available over the Agentcities network as an agent service. In order to use the service agents must employ the IBundler ontology, particularly designed to offer all its expressiveness, and the FIPA ACL. Finally, IBundler is also being integrated as a service in the framework of the SWS project, an EC-funded project (IST-2001-37134) accessible at http://swws.semanticweb.org. SWWS is intended to provide a semantic-based web service platform that shall serve as a basis for a B2C case study and a B2B case study. IBundler is to be part of the B2B case study focusing on the negotiation of services among enterprises.

On the Integration of Restaurant Services

Institut d'Informàtica i Aplicacions
Agents Research Laboratory
Universitat de Girona
Miquel Montaner, Beatriz López, Esteve del Acebo, Silvana Aciar and Isabel Cuevas
{mmontane,blopez}@eia.udg.es, acebo@ima.udg.es, {saciar,icuevas}@eia.udg.es

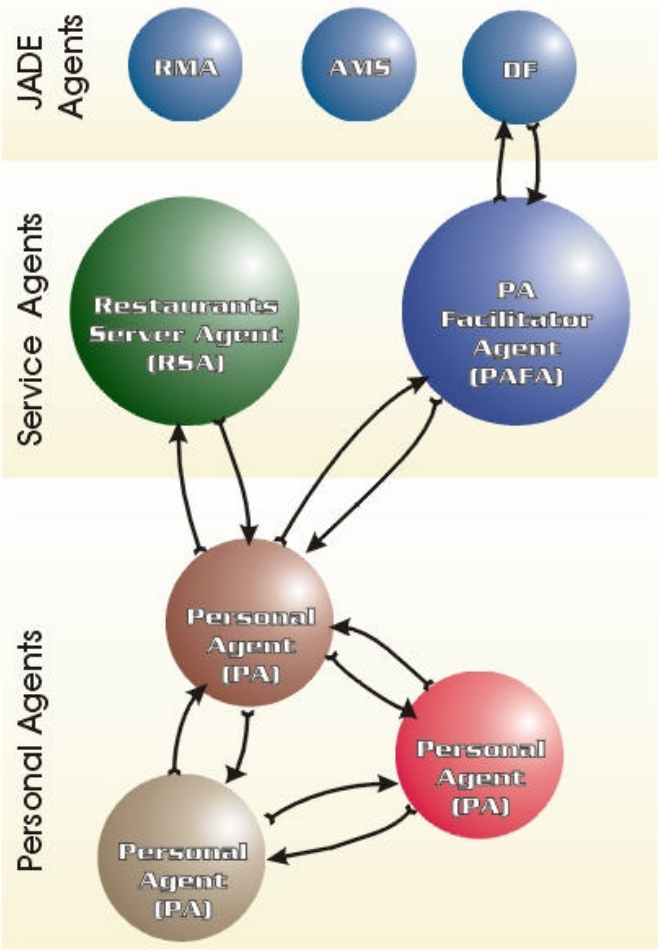
Abstract
Our contribution to AgentCities is a restaurant recommender service that consists of a multi-agent system of service agents and personal agents. The service agents offer information about restaurants and the personal agents. The personal agents provide restaurant recommendations to their users based on information on restaurants and interaction with other *friendly* personal agents. To achieve this purpose, the personal agents interact with the restaurant server agent to find information about the restaurants, with the personal agent facilitator agent to find out other personal agents in the system, and with other personal agents to find similar users and benefit from their opinions and advice. In order to improve the performance of the personal agents, case-based reasoning and trust techniques have been applied.

System Description

Recommender agents make recommendations to users according to the information available. Such information includes data on items (e.g., restaurants) as well as different profiles of other users on the system.

Many recommender agents have been developed on the Internet [4] where the typical architecture is a centralized server where users are connected in order to obtain their recommendations. However, when we apply the agents' theory to recommender systems, a standard centralized recommender system becomes a distributed world of recommender agents. Our contribution to AgentCities is in this latter line: a restaurant recommender service designed as a multi-agent system of recommender agents.

In our architecture (see figure) we can group agents into service agents and personal agents (PA). Among the service agents, we distinguish the restaurant server agent (RSA) and



the personal agent facilitator agent (PAFA).

The RSA contains all the information about restaurants. There could be many RSAs, each one having information about restaurants of a given area (e.g., a city, a county, a province, a region or a country). For example, an RSA could offer information about the restaurants in Barcelona city, in the province of Barcelona or in Spain. We actually discourage RSAs that offer information about a whole country because the agent would receive too much information. Moreover, if we design RSAs that give information about a city, we are excluding some popular restaurants situated near that city. Therefore, we advise RSAs that cover a city and its outskirts.

The RSA can provide information about any given restaurant found by name, by address or by location. It can also provide information from its case-based reasoning engine developed over the database. Due to the similarity functions among restaurant attributes, we can retrieve restaurants that are very similar to a given one, restaurants very different to a given one or restaurants closest to what the user has in mind.

The PAFA acts as a broker agent: it is in charge of assisting PAs to find and contact other PAs. If the system has many PAs, it should contain several PAFAs in order to prevent overloads. Mainly, the PAFA informs the personal agents about whether a given PA is online before contacting him/her, about the agent identification (AID) of a certain PA in order to contact him/her and about the AID of other PAs in the system in order to make new *friends*.

PAs are in charge of recommending restaurants to the user. Each user has their own PA in their local machine and the service agents placed in the server give assistance to them. Each personal agent encapsulates the user profile and filters recommendations based on this profile. Therefore, the privacy of the personal data is protected. The content-based filtering method is applied to make recommendations to the user based on the content of the items. To be specific, a case-based reasoning approach is implemented [2]. However, collaborative filtering cannot be applied since the user profiles are protected by each recommender agent and it is impossible to make direct comparisons among them. In order to take advantage of the collaborative world, we have implemented a new information filtering method based on a model of trust: the opinion-based filtering method [1,3]. The main idea is to consider other agents as personal entities which may or may not be reliable. Reliability is expressed through a trust value with which each agent labels its neighbours. The trust value is initially calculated through interaction, following a proactive *playing agents'* procedure [3]. Once the agent has received a set of *friends*, it can use them to filter information. When the agent is not sure about a recommendation or discovers a new item, it asks the reliable agents for their opinion and uses their trust values to decide whether the item is interesting for the user or not. Once the agent has the opinion of other agents, a consensus is achieved through the use of an aggregation measure. The result of the consensus provides a confidence value upon which the agent can decide on the convenience of recommending an item to the user.

We suppose that similar agents will provide pertinent opinions, but they may also provide inadequate ones. The trust, therefore, should be modified according to the results of the recommendations, in order to improve acquaintance among agents.

It is important to note that the opinion-based approach is combined with content-based and collaborative filtering methods. The performance of the content-based method is the same, but, due to trust, we are able to apply a collaborative method and improve it in two ways. Firstly, we are not looking for private data of other users but for an opinion; secondly, agents only believe in the recommendations of agents with a high trust value.

In order for our agents to communicate, an ontology has been developed. The ontology encompasses the communication between the PA and the RSA, between the PA and the PAFA and among the PAs.

The proposed system has been implemented as a prototype. In order to test its performance, we installed twenty-two agents for people in our Department. We only implemented one RSA with the restaurants of Girona city and its outskirts and one PAFA. The system was working for two months and, at present, we are starting to analyse the results obtained.

Support / Acknowledgments

This research has been developed within the IRES Project supported by the grant ACNET.02.50 and the DAF-DAF Project supported by the CICYT grant DPI2001-2094-C03-01.

References

1. M. Montaner, B. López, and J. L. de la Rosa. Developing trust in recommender agents. *In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). Cristiano Castelfranchi and W. Lewis Johnson (Eds). ACM Press. vol. 1, pp. 304-305. Bologna (Italy)*, 2002.
2. M. Montaner, B. López, and J. L. de la Rosa. Improving Case Representation and Case Base Maintenance in Recommender Agents. *In Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR'02). Susan Craw, Alun Preece (Eds.), Lecture Notes in AI N2416. Springer-Verlag. pp. 234-248. Aberdeen (Scotland)*, 2002.
3. M. Montaner, B. López, and J. L. de la Rosa. Opinion-Based Filtering Through Trust. *In Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA'02). Matthias Klusch, Sascha Ossowski and Onn Shehory (Eds.), Lecture Notes in AI N2446. Springer-Verlag Berlin Heidelberg, pp. 164-178, Madrid (Spain)*, 2002.
4. M. Montaner, B. López, and J. L. de la Rosa. A Taxonomy of Recommender Agents on the Internet. *To be published in Artificial Intelligence Review*, 2003.

A5: MAS-VPR

MAS Vehicle Routing Problem with Time Windows

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science
Oleg Karsaev
ok@mail.iias.spb.su

Abstract
The entry is auction-based multi agent system for Vehicle Routing Problem with Time Windows (VRPTW). The application was implemented to solve a well-known benchmark developed by Solomon. This benchmark incorporates a shipping company possessing a depot and a number of trucks and a set of orders from different customers. Given constraints the task is to allocate orders over trucks. At that number of trucks and total distance covered by them should be minimal. The system to be demonstrated includes *Shipping company* agent playing the role of auctioneer and several *Truck* agents playing the roles of contractors. Specification of the application developed is represented in a specific language that is supposed by the MAS development tool used. This tool also used for code generation and deployment the application in a local network. Applied MAS is provided for user interface that allows visualizing results of auctions and supports interactive mode of task solving.

Problem description

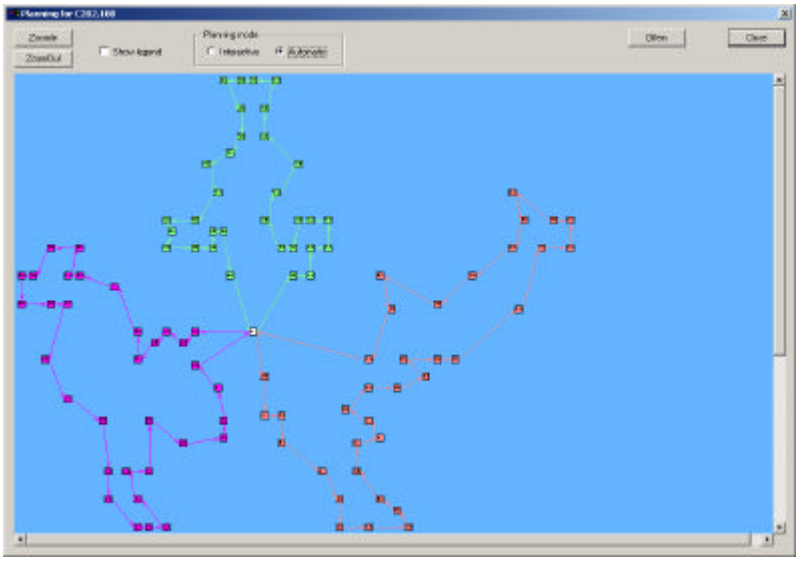
A shipping company has to carry out a "portfolio" of orders to deliver the loads from depot to customers. An order specification comprises coordinates of customer's location, load volume, service time duration and service-time window. Every customer must be serviced within predefined service-time window. If a vehicle reaches a customer before begin of service-time, a waiting time occurs. The delivery time spent by vehicle is a sum of the waiting time, service time and travel time. Travel time between a pair of customers is equal to the distance between them. The objective of VRPTW is to design a schedule that provides service for all customers while minimizing the number of vehicles, the total travel distance passed by all vehicles without violating vehicle capacity constraints and customers' time-service windows.

General description of multi agent system

Multi agent system (MAS) consists of a *Shipping company* agent playing a role of auctioneer and several *Truck* agents playing the roles of contractors. Shipping company agent receives "portfolio" of orders to allocate them over Truck agents. Orders allocation and delivery scheduling is performed on the basis of an auction protocol. In this application the agents' behavior is non-competitive. For each order at hand, Shipping company agent selects winner on the basis of a criteria, which is changed from step to step of the order allocation according to a strategy. In general case the order allocation process is split into two stages. At the first stage, an initial allocation of the orders is computed. At the second stage a subset of allocated orders that are assessed as allocated not in the best way is reallocated. The last stage can be repeatable.

In progress of allocation procedure, Shipping company agent provides visualization of the customers' and depot locations and also current routes of trucks. MAS can operate in both *automatic* and *interactive* modes. In the last case user's intervention in allocation process is permitted.

Input data for system testing



The Solomon's benchmark data sets developed for VRPTW task are stored in the MAS database. User performing MAS testing is provided a special interface that makes it possible to select benchmark data set manually. Each such a data set specifies 100 orders, but user is provided a capability to select any its subset. The benchmark data sets are additionally labeled by C, R and RC. In data sets labeled by C customer locations are clustered, in data sets labeled by R customers' locations are random, and in data sets labeled by RC customers' locations are mixed. The data sets labeled by index 1** have "short scheduling horizon", and the data set labeled by index 2** have "long scheduling horizon". While testing MAS, for each benchmark data set it is necessary to assign the maximal number of truck agents and strategy of agents' behavior.

First stage – initial allocation of orders

The first step after MAS start up, the user selects manually testing data. Afterwards, Shipping company agent opens user interface window that visualizes geographical locations of the customers and depot. In progress of the order allocation, user interface visualizes current statuses of truck routes. At any time, user is also provided for ability to intervene in solving procedure and use interactive mode of MAS operation. In interactive mode user is provided for more detailed information about current status of the solving procedure and can revise decisions made automatically by Shipping company agent.

Scenario of each auction is as follows. Having selected the next order for bargaining, the Shipping company agent determines the Truck agents that would be able to propose a good offer and sends them the order specification. Each truck agent received order specification calculates the "cost" of the order execution and some additional parameters. The cost is calculated as an increase of the current route distance provided that the proposed order is included in its route. After that the Shipping company agent analyses the order execution costs of bargaining truck agents and also some other parameters, and select the winner on the basis of a criteria that is chosen according to the accepted strategy and can be different for each step of allocation process. If the interactive mode is implemented user can visually analyze the current routes of trucks and the order under allocating and revise the winner selected by Shipping company agent. At that user can also consider a possibility to manually allocate unallocated orders. Interactive mode of MAS operation helps to user to learn new heuristics.

Second stage –reallocation of an order subset

The objective of the second stage is to improve quality of generated schedule. At the beginning of the stage each Truck agent forms two subsets of the orders: the orders which "agent would like to buy" and the orders which "agent would like to sell". A motivation of this "agent's desire" is as follows. The cost of order execution depends on the route formed to the current step. If the truck was allocated a new order, the current route of the truck would change and the costs of the

already allocated orders would change accordingly. This is why a revision of the allocations made can be profitable.

To select the above "desired" and "undesired" sets of order, the Truck agents use a decision making rule according to an earlier selected behavior strategy. These sets of orders are sent to Shipping company agent for reallocation. In the interactive mode the set of orders can be revised by user. At that user can eliminate some of the offered orders and/or add new orders for reallocation based on visual analysis of current routes.

Reallocation of selected orders is performed by the same way as on the initial stage, but according to special sequencing. At first the orders selected "for buying" are allocated, afterwards orders added by user are allocated and finally the orders "selected for sale" are allocated. In interactive mode sequencing of order reallocation can be revised by user.

The second stage can be repeated several times while this steps result in improvement plan's quality.

Who may be user of the system and an example usage scenario?

Transportation companies may be considered as potential users of the system. Of course, the benchmark describes problem domain not in depth. Nevertheless the problem statement can be considered as a starting point to go to a more real-life constraints and peculiarities.

Brief description of organization

St. Petersburg Institute for Informatics and Automation is governmental research institution. It is one of the leading institutions of Russia in the area of theoretical and applied Computer Science, Artificial Intelligence and Computer Systems. Other areas of research are image processing, information technologies for ecology, telemedicine, environmental sciences, etc. The Institute scientific stuff consists of about 150 researchers. Between them, more than 60 have Ph.D degree.

A6: Settler

Agent-based Settler Game

University of Hamburg, Faculty of Informatics,
Team: Michael Duvigneau, Michael Köhler, Daniel Moldt, Christine Reese, Heiko Rölke
moldt@informatik.uni-hamburg.de

Abstract
In the context of agent oriented software development we use a settler game to illustrate our approach to architecture, modelling, and implementation. The approach is aiming at open agent systems. The development is based on a distributed and truly concurrent framework including Reference nets and Mulan illustrated below. Settler itself is a multi-player game with a lot of possible interactions between agents.

Application description

"Settlers of Catan" is a game. This game gave the initial motivation for the presented implementation. Originally it is played with three to six players on a board. It is round-based, the players can interact freely by buying (event cards), building (settlements, roads, villages), trading (at ports, with the bank), negotiating (with each other about materials for building and buying), and by cooperating ("if you build over there, then I will stay here"). A robber can be placed tactically on the board and event cards can be used to influence the situation. The basic way of getting resources (Lumber, Bricks, Ore, Grain, and Sheep) is to harvest at the beginning of each round, depending on the roll of two dices. The expansion through building is limited by the coast of the Catan island.

The rules of the game allow a good deal of inspiration, to try the whole range of agents and platforms possibilities. For the base game only basic abilities are needed: Starting a conversation and responding to messages using protocols. Possible extensions to the rules could use socially motivated interactions between player agents, or mobility. *Social inter-actions* would need an elaborated planning function and protocols. A restricted *mobility* is implemented and provided by the platform: The code is transferred implicitly (only the names of the protocols), and a part of the dynamic state is transferred (the knowledge base), whereby no information is lost (when the agent is idle).

The basic game is implemented using an agent-oriented development approach. Agents represent the players and some central roles in the game, as the board and the game-control.

The Underlying Framework

The technical environment for the development consists of five layers: Java, Renew, Mulan, Capa, and Ace.

- *Renew* ("Reference Net Workshop") is an efficient Petri net editor and simulator. Petri nets are a basic formalism to model concurrent systems. Reference nets are a high-level Petri net formalism provided by Renew. Due to their operational semantics it is possible to run them as a program. Very naturally they combine the concept of true concurrency with visual programming. Within Renew, it is possible to link Java code to a transition (the basic acting element of a Petri net). This allows to smoothly integrate any Java code.

- *Mulan* (“Multi Agent Nets”) is our conceptual agent framework. The concurrency of the underlying formalism is a big advantage for the design of multi agent systems. Mulan’s architecture consists of four levels: multi agent system, platform, agent and protocol level. When developing a new agent based application or service, the knowledge base and the set of protocols are designed for each agent.

- *Capa* (“Concurrent Agent Platform Architecture”) implements the communication architecture according to the FIPA specifications. Included are obligatory services like AMS, DF, and a central transport service. The platform itself is implemented as an agent with special possibilities. This adds diversity in FIPA-compliant platform implementations and architectures.

- *Ace* (“Agentcities Environment”) will connect the framework to Agentcities. This includes a ping agent, and adopting of naming conventions for agents. With Ace, Settler can be seen as a service within Agentcities.

Usage scenario

Michael and Julia start a game with three additional computer players. In the startup round, the five players put their initial settlements and roads on the board (the human players use the mouse). The GUI shows the game board, the current resources of a player and the incoming messages. With buttons it is possible to send messages to other agents involved in the game.

During the game, Michael wants to build a settlement. He sends an action request to the “island”-agent. This message contains what and where he wants to build. Agent “island” sends an action request to the “bank” agent to freeze the resources. “island” also reserves the place on the game board. If any of these fails, the reserved resources are freed, otherwise the resources are removed and “island” sends a “done”-message to Michael.

Implementation status

A first proprietary but stable version of Settler has been built: version 1 of Settler is in use by students. This is currently completely redesigned and re-implemented as version 2, which is FIPA-compliant and has further elaborated features. Extensions are being implemented for Agentcities (Ace). If Ace happens to provide basic service until february, the agents of Settler will be visible within Agentcities.

Settler is meant as a prototype, mainly for research on agent based application development and planning algorithms (adaptive system development). Settler includes four agents that make up the game: game control, island, bank, and marketing place. One or more player agents are possible, each with GUI or with planning capability. It is possible to play Settler distributed on several computers. Human and agent players can build and harvest.

Several important features of agent systems like concurrency, distribution, encapsulation, and autonomy can be seen in the implementation. The execution environment (Renew) allows inspection of static and dynamic Petri nets and Java token objects, and the Mulan inspection tool enables easy access to protocols, knowledge bases, and messages.

There is no simple agent access designed for the various services provided and used in the game. These are tightly embedded in the interactions. Settler will use other agent’s services rather than being used by other agents. Anyway, it is possible to implement your own player agent to join the game.

Bibliography

- M. Duvigneau, D. Moldt, and H. Rölke. Concurrent architecture for a multi-agent platform. In *Proceedings of the 2002 Workshop on Agent Oriented Software Engineering (AOSE'02),* 2002.
- M. Köhler, D. Moldt, and H. Rölke: Modelling the behaviour of Petri net agents. In J.M. Colom and M. Koutny, editors, Proceedings of the 22nd Conference on Application and Theory of Petri nets, volume 2075 of LNCS, pp. 224-241, Springer 2001.
- O. Kummer: Introduction to Petri nets and Reference Nets. Sozionik aktuell, Nr. 1, 2001. ISSN 1617-2477. Available at http://www.sozionik-aktuell.de .
- O. Kummer, F. Wienberg and M. Duvigneau: Reference Net Workshop (Renew). Universität Hamburg 2001. Available at http://www.renew.de.

An Travel Market Framework in Agentcities

UMBC-TAC, U.Maryland Baltimore County
yzou1@cs.umbc.edu

Abstract
Travel Agent Game in Agentcities (TAGA) is a framework and infrastructure that extends and enhances the original Trading Agent Competition (TAC) system [Wellman 99] to work in the Agentcities environment. Auction service is added to enrich the Agentcities service feature. RDF and WSDL are used to enhance the interoperability of FIPA compliant agent and web service. The power of Agentcities technology plays a big role in building this distributed and open market framework, which further facilitates the web-based e-commerce research.

Overview

TAGA game simulates the real world travel market with hybrid mechanisms buying direct from service providers or participating the auctions. Six kinds of agents play different roles and collaborate to help customers organizing travel itinerary. Multiple travel agents compete with each other, trying to acquire more customers and maximize the profit. TAGA can be used in research of travel market and auction strategy. Running on FIPA-compliant platform and Agentcities.net, TAGA can also be used as the test-bed for agent interaction in the distributed and open environment.

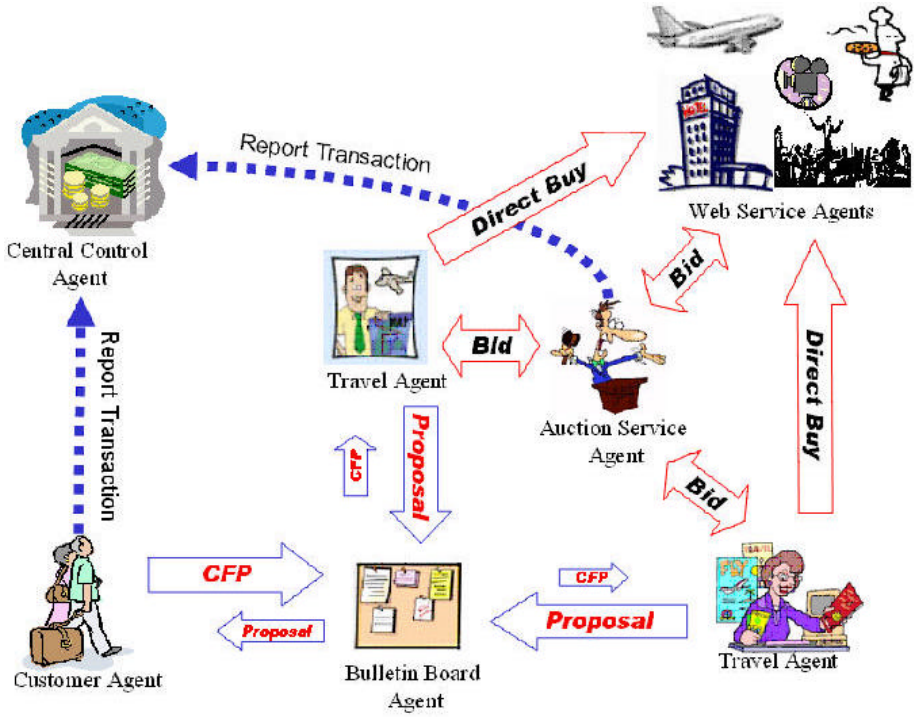


Figure 1: the agents and communications in TAGA

- *Customer Agent* – represents customers that have preference values;
- *Travel Agent* – organizes travel itinerary for customers and interacts with service provider;
- *Bulletin Board Agent* –sends customer information to subscribed travel agent;
- *Web Service Agent* – models an actual representation of the travel service
- *Auction Service Agent* - provides auction service
- *Central Control Agent* –controls the game and audits all transactions

Game Rules

The customers travel from City A to resort R and spend several days there. A travel itinerary includes a round-trip flight and corresponding hotel accommodation. Each entrant to the game is a travel agent. To win the game, the travel agent needs to develop best buying and auction strategies and maximize the profit by signing contract with customers and buying/bidding the limited travel services resource from the travel service provider.

TAGA Highlights

(1) Auction services in Agentcities

New Auction Services – Inspired by FIPA specifications[2] and current web auction sites[3], TAGA proposes two new auction types: Web English Auction and Web Dutch Auction. They allow new bidders to enter auction freely before the auction close and end the auction when reach expected close-time.

(2) Open System

Multiple FIPA platforms – Both AAP and JADE platform are used to develop TAGA agent. The agents support HTTP and SOAP message transport interface, can be started from the web by using SUN's Java Web Start technology.

Web Service Interface – Web services agent in TAGA provide the WSDL profiles and SOAP message interface to support SOAP-based web service access.

(3) Flexible Market Framework

True Agent implementation – All agents in TAGA are independent agents and act on their own behalf.

Multiple purchase options – TAGA supports both *Direct Buy* and *Auctions* purchase method. The travel agent can choose the purchase methods as well as the service providers.

(4) RDF as Content language

RDF is chosen as ACL content language because: (a) it is easy to parse and more readable; (b) it supports knowledge representation for service description, customer preference, and reputation, allowing inference on such information; (c) RDF is widely used in web service.

(5) Compare with TAC

TAC used centralized auctions as the sole mechanism for service discovery, communication, coordination, commitment, and control among the participating software agents, while TAGA supports hybrid purchase mechanisms and has more trade off for the agent strategy. By simulating real world travel market, TAGA is not only the test-bed for market strategy, but also the test-bed for agents and web services.

About us

TAGA game is developed by UMBCTac team. Student team members include Youyong Zou, Li Ding, Harry Chen and Rong Pan. Faculty advisor is Dr. Finin. UMBCTac team joined the past three-year's TAC games, ranking 4th in both 2000 and 2002.

Reference

- [Wellman 99] Michael P. Wellman and Peter R. Wurman. A trading agent competition for the research community. *IJCAI-99 Workshop on Agent-Mediated Electronic Commerce*, Stockholm, 1999

[2] FIPA XC00031(English auction), FIPA XC00032 (Dutch Auction).
[3] http://ebay.com/, http://www.priceline.com/.

I1: EFA

Embedded FIPA Agents

Dynamic Interaction group
Media Lab Europe
Jamie Lawrence
jamiel@mle.media.mit.edu

Abstract
This project aims to deploy agents on a self-contained, FIPA-compliant agent platform hosted by an embedded device. The aim is to allow agents to be directly embedded into physical spaces and objects in a manner not supported by the typical beige-box PCs.

Fraggle Rock is the smallest Agentcities platform currently deployed, measuring just 3x2.5" and running a modified version of the JADE/JADE-LEAP platforms. It remains FIPA-compliant and includes a number of innovations important to open, dynamic environments.

Brief Description

The goal of this research is to investigate emergent phenomena in an ad-hoc network of wirelessly connected agent platforms. The field of ad-hoc networking provides a testbed for the multi-agent paradigm's ability to provide seamless interoperability between heterogeneous services. This environment is categorised by the random, transient interactions between nearby devices and the challenge is to build applications that can discover, recognise and successfully interact with other nodes. To do this we need to move multi-agent systems into small physical and computational spaces, ensuring that they can discover and communicate with each other in a reliable and maintenance-free manner.

As a necessary precursor to achieving this goal, we have developed Fraggle Rock, an embedded, self-contained, FIPA-compliant agent platform running on a Systronix JStik. The JStik is a 3x2.5" board built around an aJile aj-100 native Java processor, 2MB SRAM, 4MB Flash RAM, 2 x RS-232 ports and a 10BaseT Ethernet port. The processor supports a restricted subset of Java called the Connected Limited Device Configuration (CLDC) of the Java 2 Micro Edition (J2ME). The processor executes native Java bytecode and so removes the need for an operating system or Java virtual machine, which in turn produces far better performance than other CLDC devices (mostly smartphones and low-end PDAs). The JStik has more connectivity, processing power and memory than our envisioned ad-hoc node but it will allow us to fix the majority of problems in a stable networked environment before optimising the platform onto the smaller JStamp-based wireless module.

The agent platform is composed of parts from JADE[i], JADE-LEAP[ii] and a number of new components. In order to support an ad-hoc network it is necessary to remove any dependencies on established infrastructure. In the case of the JADE-LEAP platform, this has meant removing the distributed container support and moving the FIPA-compliant parsers and agents from the main container onto a CLDC device. Components such as agent mobility, which couldn't work on CLDC, were removed along with other unnecessary components such as tool and wrapper support (for supporting access from other Java programs). New components were developed to support platform discovery, leasing and a FIPA-compliant HTTP transport protocol. In addition,

the Ontology Bean Generator for JADE[iii] was modified to output CLDC compatible ontology classes from a Protégé model.

From the perspective of the agent developer, Fraggle Rock still supports most of the functionality found in JADE including the behaviour scheduling model, interaction protocols, content language and ontology support. Agents written for JADE can run on Fraggle Rock with only minor modifications (mostly related to the differences between J2SE and CLDC).

Applications

Fraggle Rock enables applications that PC-based agent platforms don't. The small board can provide weather readings in a high-level, agent-edible format from a remote weather station. The platform could be embedded into an LCD screen to provide a "Life Monitor" for displaying news, weather, bank balances, and other information retrieved from agents on the Internet. Optimised for a smaller wireless node, the platform could be used to support a smart conference badge or the wireless distribution of weblog entries. It can be used as a "smarter" node within a distributed sensor network to manage the activation of smaller sensors, analyse the sensor data within the network, and take the necessary actions.

Conclusions

Fraggle Rock is the smallest node currently deployed in the Agentcities network and yet remains interoperable with other FIPA-compliant platforms around the world. It incorporates features that could be used to improve the network architecture of Agentcities.

This research prototype is a stepping-stone on the way to deploying agents on even smaller devices in a wireless ad-hoc environment. In particular, the goal is to reduce the physical size of the platform by half, utilise a low-power radio for communication and optimise the platform for tighter processor and memory constraints.

The platform is currently undergoing testing live on the Agentcities network and a number of demonstrable applications are under development.
Further information about this project can be found at http://sprocket.mle.ie.

About Media Lab Europe

Leveraging the innovative and entrepreneurial operating model of the world-renowned MIT Media Lab, Media Lab Europe adopts an interdisciplinary approach to researching the ways in which new technologies expand human potential. Media Lab Europe's home is in Dublin where it was established two years ago as a 10-year joint venture between the MIT Media Lab and the Irish Government. 100 people work for the not-for-profit research institute, across seven groups and supporting areas. To find out more about Media Lab Europe please visit http://www.medialabeurope.org.

I2: Entish

eLingua for service description and composition

enTish Team
Institute of Computer Science
Polish Academy of Sciences
sambrosz@ipipan.waw.pl

Abstract
A new technology for service description and composition in open and distributed environment is proposed. The technology consists of description language (called Entish) and composition protocol called entish 1.0. They are based on software agent paradigm. The language and the protocol are merely specifications. To prove that the technology does work, the prototype implementation is provided. It is still under testing and is available for use and evaluation via www.ipipan.waw.pl/mas/. The specifications were created on the basis of the requirements produced by the Service Description and Composition Working Group within the framework of Agentcities.NET. Related work was done by WSDL+BPEL4WS+(WS-Coordination)+(WS-Transactions), and DAML-S.

Description

A new technology for service description and composition in open and distributed environment is proposed. The technology consists of description language (called Entish) and composition protocol called entish 1.0. They are based on software agent paradigm. The language and the protocol are merely specifications. To prove that the technology does work, the prototype implementation is provided. It is still under testing and is available for use and evaluation via www.ipipan.waw.pl/mas/ . The specifications were created on the basis of the requirements produced by the Service Description and Composition Working Group within the framework of Agentcities.NET. Related work was done by WSDL+BPEL4WS+(WS-Coordination)+(WS-Transactions), and DAML-S.

The language Entish is fully declarative contrary to BPEL4WS and DAML-S. A task (expressed in Entish) describes the desired static situation to be realized by the composition protocol. Hence, our technology for realizing the idea of Web services is different than the one proposed by IBM and Microsoft. The complete documentation of our technology is available on request from sambrosz@ipipan.waw.pl, and is also available at www.ipipan.waw.pl/mas/

The prototype system is available for use and evaluation via three www interfaces. The first interface called *EntishDictionary* serves for introducing names for new data types, their attributes, and new functions to the language Entish. The second interface called *serviceAPI* is for joining applications (that implement the new functions) as networked services to our system. The third interface called *TaskManager* is devoted for a client to specify its task in Entish, and provide initial resources the for the task realization. Hence, from the outside, i.e., from service providers and clients point of view, the system consists of the three interfaces: EntishDictionary, TaskManager, serviceAPI. What is inside, that is, the system engine is transparent for the providers and clients. The engine implements the language and the protocol. The protocol realizes the clients' tasks by discovery of appropriate services, their composition and invocation. The www interfaces are user friendly so that to use the system almost no knowledge on XML, Entish syntax, and entish 1.0 is needed.

EntishDictionary (ED for short) serves also as an ordinary dictionary, i.e., for looking at the existing ontologies as well as for explanations of names used in the language. Ontology is meant as a collection of names of resource types, their attributes, and functions defined on these types. It is supposed that they come from one application domain. Any user can introduce its own ontology. There is no conflict with names, because short names introduced by users are automatically extended to long names that are URIs. ED has also additional functionalities. It allows a service provider to create Entish formula that describes operation type performed by its application. The formula along with some additional information about the host on which the application is running is sent automatically to the system (actually to *serviceServer*) for registration and publication.

The second functionality of ED is that it allows a user to create task; usually it is a composition of functions. The task is sent automatically to TaskManager for realization. It is important to note that task is merely an abstract description of what is to be realized, so that it does not indicate what services could realize it.
Note that EntishDictionary realizes some functionality of UDDI concerning service classification. However, service registration and publication is done inside the system.

TaskManager is a GUI that, given a task from a user, creates and manages appropriate interfaces for delivering (by the user) initial resources needed for the task realization. Then, the task is sent to the system (actually to *agentServer*) for realization. If the system is ready to realize the task (it means that a workflow consisting of appropriate services has been already constructed), the TaskManager asks the user to provide the initial resources according to the constrains returned by the system.

The interface serviceAPI provides Java classes and explanation for creating service (according to our architecture) by a service provider.

Implementation / Use status

The implemented system is a prototype still under testing. The purpose of the prototype was to show that the description language and the composition protocol constitute a working technology. The prototype serves to gather experiences needed to improve the language and the protocol.

Relevance to open agent systems

Please note that the only requirement for joining heterogeneous agent system to our infrastructure is that an agent should speak Entish if it wants to realize a task. If an agent provides a service, it should only implement appropriate interface (serviceAPI) to register its service to the infrastructure.

Relevance to Agentcities

The Agentcities objective is to construct a worldwide, open infrastructure (middleware) for on-line systems hosting diverse agent based services which can interact with one another dynamically, autonomously and intelligently to form new compound services. Our technology may be considered as a proposal for providing such open infrastructure. The technology is the realization of the requirements produced by the Service Description and Composition Working Group of Agentcities.NET.

I3: ORCAS

Open, Reusable and Configurable multi-Agent Systems

IIIA-CSIC/IBROW
Mario Gómez
mario@iiia.csic.es

Abstract
ORCAS is an electronic institution designed to configure multi-agent systems on-demand, by using a knowledge modelling approach to describe agent capabilities. Specifically, we propose Problem-Solving Methods as the paradigm to describe and reason about agent capabilities. The configuration of a MAS is performed in two steps: capability-brokering, which goal is to find a configuration of agent capabilities satisfying the requirements of a problem, and team-formation, that uses the configuration of agent capabilities to form a team of agents instructed to solve that problem in a cooperative way.

Description

ORCAS is an institutional framework that provides the infrastructure required to configure a MAS on demand, for each request to solve a problem, according to its requirements. ORCAS has been modelled and implemented as an electronic institution where requesters and providers of capabilities meet. The electronic institution provides a shared social knowledge required by both requesters and providers to achieve a successful interaction. Furthermore, interactions are mediated by institutional agents that provide an added value to both requesters and providers. The mediation services are those required to form teams of agents capable of solving a problem according to its requirements.

The added value is twofold: for the requesters, it avoids them of finding and selecting the providers suitable for solving their problems; and for the providers, the institution supports and facilitates interaction and cooperation among them in order to achieve complex problems that capability provider-agents cannot solve as isolated agents. Configuration of a MAS in ORCAS requires that both agent capabilities and problem requirements be specified using the same *Agent Capability Description Language (ACDL***)**. The ACDL used in our institution is based on the *Problem-Solving Methods (PSM*) paradigm, which basic idea is to declaratively describe software

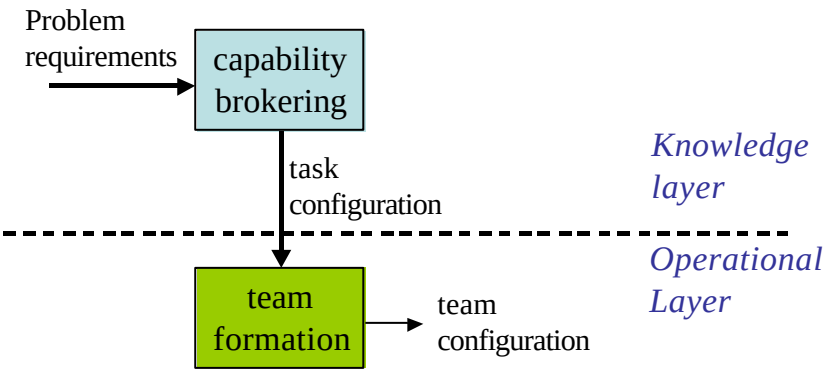


Figure 2: Two layers MAS configuration

systems by decoupling tasks, PSMs and domain models, and all these components can be described with independent ontologies in order to facilitate reuse. We describe a MAS using these three classes of components: *Tasks* describe which kind of problems agents can solve, *PSMs* describe specific capabilities to solve a task, and the *domain models* describe the domain knowledge required by tasks and PSMs. We have modelled the configuration of a MAS as having two layers: the *knowledge layer* and the *operational layer*.

(1) Configuration at the *knowledge layer* refers to the process of finding a configuration of agent capabilities satisfying the requirements of a problem. We call this process **capability brokering**. The result of capability brokering when using PSMs and tasks to describe agent capabilities is a **task configuration**: A hierarchical tree that decomposes a task into a subtasks, where each task has been assigned with a PSM suitable for that task, and optionally some domain model and knowledge.

(2) Configuration at the *operational layer* refers to the process of forming and instructing a team of agents to solve a problem in a cooperative way, according to a task configuration. This process is called **team formation**, and is divided in two stages: *task allocation* and *team instruction*. *Task allocation* is the process of selecting the members of the team and deciding which subtasks will solve each member, while *team instruction* is the process of informing team members about its roles in the team and the social knowledge they require to cooperate with the other team mates.

The different agent roles in the ORCAS e-institution are the following:

(1) *Providers*: agents providing certain capabilities for solving problems. We call them **Problem-Solving Agents** (PSA), as their capabilities are described as Problem-Solving Methods.

(2) *Requesters*: human or software agents requesting to solve a problem. We have included the **Personal Assistant** (PA), an agent acting on behalf of a human user. The PA is able to specify problems in terms understood by the institutional agents, freeing users of knowing the technical details.

(3) *Middle (institutional) agents*: agents mediating between service requesters and providers. These agents are responsible of finding, selecting and instructing PSAs to solve a problem in a cooperative way. We consider three classes of middle agents: **Librarians**, **Capability-Brokers** and **Team-Brokers**. Librarians are dynamic repositories of agent capabilities; capability-brokers are able to obtain a configuration of agent capabilities, and team-brokers deals with the formation of agent teams.

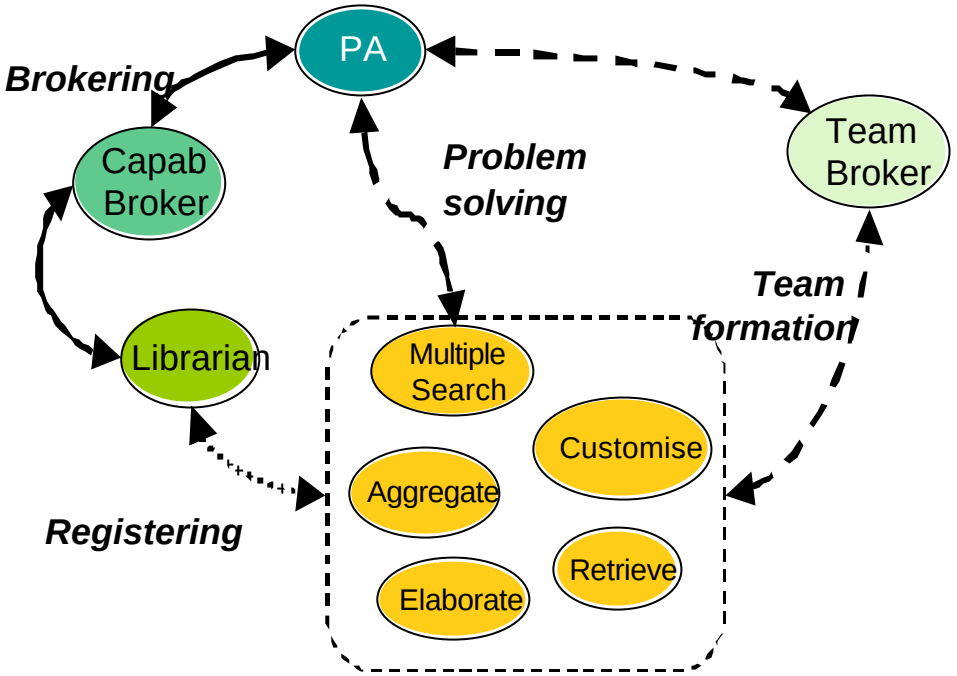


Figure 3: Multi-agent architecture

Figure 3 shows an overview of the multi-agent architecture of an ORCAS e-institution, where you can also see the main interactions in which agents are involved. These interactions are specified as communication scenes, including the following:

1. *Registering*: within this scene the PSAs register/deregister its capabilities to the librarian, keeping the library updated.

2. *Capability brokering*: this scene describes the pattern of interaction needed to obtain a configuration of agent capabilities. The participants are the PA, the capability-broker, responsible of finding the configuration, and the librarian, holding the description of the available agent capabilities.

3. *Team formation*: this scene describes the interaction protocol for selecting and instructing the members of a team according to the capability-configuration. The outcome of the scene is a team-configuration. The agents involved are the PA, the available problem-solving agents, and the team-broker, responsible of selecting and instructing the team members.

4. *Problem-solving* scene: once a team of agents has been formed and instructed to solve a problem, the team-mates go to this scene where they execute the selected PSMs to solve the problem, cooperating with other team-mates according to the team-configuration. The participants are the PA, holding the data of the problem, and the members of the team.

Application Example

ORCAS has been used to build WIM, an application to look for medical bibliography in Internet. WIM includes a collection of information agents and some domain models in medicine, like a medical thesaurus. The user can specify queries and preferences, and its personal assistant will transform them into a problem specification (data and requirements) that is used to customize a

new team of agents for that problem. WIM can be accessed from a web form or from a FIPA compliant agent using a request-inform protocol.

Summary of contributions

Electronic institutions achieve a trade-off between autonomy (local control) against centralization (global control). The ORCAS e-institution in particular is designed to support configuration of multi-agent systems in open environments, where providers of services can register or deregister its capabilities dynamically. Our framework is focused on agent interaction, thus it is neutral about the internal architecture of agents, allowing for a multiplicity of agent architectures and platforms, whenever they follow the interaction protocols and the rules specified by the institution. ORCAS is designed to facilitate agent interactions, reducing the social knowledge required from participants, as the specialized knowledge and the reasoning skills required to configure a team of agents are handled by the institutional agents. Another issue that makes our approach well suited to open agent environments is the separation of tasks, PSMs and domain models, as this separation facilitates reuse, for example, we can reuse the WIM information agents in other application domains just providing the system with the description of the new domain models (a different thesaurus, different information sources and so on). We have adapted a software architecture for describing knowledge systems (UPML) in order to describe multi-agent systems,. Furthermore, we have developed a new configuration method called *Constructive Adaptation*, which uses CBR to constrain the search over the space of possible configurations.

Ongoing work

We are currently modifying the ORCAS framework in order to make MAS configuration more dynamic and fault tolerant, allowing for interleaved configuration and execution, load balancing techniques and dynamic replacement of teammates in case of communication or agent failures.

Acknowledgements

ORCAS has been developed under the IBROW project (http://www.swi.psy.uva.nl/projects/IBROW3/home.html) at the Artificial Intelligence Research Institute (IIIA-CSIC http://www.iiia.csic.es).

WIM: http://wim.iiia.csic.es/Wim-Demo/Demo-form.html

ORCAS: http://wim.iiia.csic.es/~mario/ORCAS.html

The Otago Ontology Repository

Department of Information Science,
University of Otago, Dunedin, New Zealand
Jin Pan, Stephen Cranefield and Dan Carter
{panji016, scranefield, dcarter}@infoscience.otago.ac.nz

Abstract
The Otago Ontology Repository is a prototype implementation of a design to allow ontology designers and agents to use open Web standards to publish and retrieve ontologies and metadata about them. Key features of the system include the use of the HTTP protocol following the REST architectural style, the representation of the recorded information about ontologies and the repository information schema using RDF and its schema language RDFS, and the use of URNs to identify ontologies.

Motivation

In order for ontologies to be shared and reused, an ontology repository is needed to store and maintain ontologies. We believe that for agent and ontology technology to gain widespread acceptance, it should be possible for agents to directly access ontologies using a well known protocol, and the information schema underlying an ontology repository should be easily accessible in a declarative format. Therefore our approach differs from the ontology service proposed by the Foundation for Intelligent Physical Agents (FIPA) [1], which requires the existence of a specialised ontology agent that mediates between other agents and knowledge representation systems storing ontologies, and the OKBC protocol [2], which specifies a procedural interface for querying knowledge representation systems. Instead we base our repository architecture on the Web standards HTTP and RDF. Also, as repositories may be moved from host to host, and/or replicated in different locations, we choose the Web standard for location-independent identifiers—Uniform Resource Names (URNs)—to identify ontologies in the repository.

FEATURES

The REST architectural style

The Representational State Transfer architectural style (REST) [3] was designed as an abstract model of the principles underlying the World Wide Web architecture. REST systems are considered to be highly scalable due to the use of stateless communication and their support for the caching of request responses (response messages should indicate whether they can be cached). Two of the key principles underlying REST are the following:

- *Resources*, identified by resource identifiers, are considered to be the fundamental unit of information. Resources are abstract notions that are not directly transferred across a network. Instead a *representation* of the current or intended state of a resource is transferred between system components as a sequence of bytes together with metadata describing those bytes. Multiple representations of a resource may be available using different *media types.*

- There should be a uniform interface for interactions between system components. When applied to HTTP-based communications on the Web, this means choosing to make use of the existing HTTP request types such as GET, POST and PUT in ways that respect their semantics (e.g. POST creates a new resource and GET retrieves one) rather than

encoding component-specific operations by including a query expression at the end of an HTTP URL.

The REST principles are applied to the repository by using an HTTP POST request to create a new ontology resource in the repository, with a representation of the ontology in some specified media type (e.g. application/rdf+xml, image/gif, text/plain) included in the body of the request. Alternative representations of the ontology in different media types can subsequently be uploaded using HTTP PUT. Ontologies are downloaded by performing an HTTP GET, with the Accept header specifying the desired media type.

Ontology metadata in RDF

An ontology repository must have some underlying model of the structure of the information that is stored in it. We use RDF Schema to describe the concepts used to represent information about ontologies. Information about particular ontologies stored in the repository is stored in RDF format. Figure 1 shows a graphical depiction of some sample RDF information about a particular ontology, the conceptualisation it represents, and its author. At present the RDF schema used addresses only information about ontologies as a whole (e.g. the name, version, author and representation language). It would be straightforward to extend this to include information about concepts defined within ontologies and the relationships between them. The schema could also be extended to allow mappings between the concepts within different ontologies to be represented.

To allow agents to assert and query information about ontologies, the recently proposed Rdftp protocol [4] is used. This provides an HTTP encoding of RDF triple assertions and queries.

Support for URNs

URNs are generated and returned to the client when ontologies are posted to the repository. These should be used for dereferencing the ontology in the future in case the repository has been moved to a different server. However, as the standard Web infrastructure does not yet include support for the dereferencing of URNs, we have provided this functionality in the following way. When an ontology is posted to the repository, it is made accessible via a particular URL on the repository server's host. The URN that is generated for the ontology is then associated with the URL by using Rdftp to assert a triple representing the mapping. To access an ontology using a URN, an agent (or a GUI client for human users) must first use Rdftp to look up the currently associated URL. If the repository is moved to a different host, the RDF triple representing the mapping can be changed. This simple approach is sufficient for the present, but a more sophisticated mechanism involving a hierarchy of mapping databases will be required in the future to allow for replicated repositories. The URN resolution mechanism has been wrapped up as a Java Agent Services (JAS) service in Otago's OPAL agent platform.

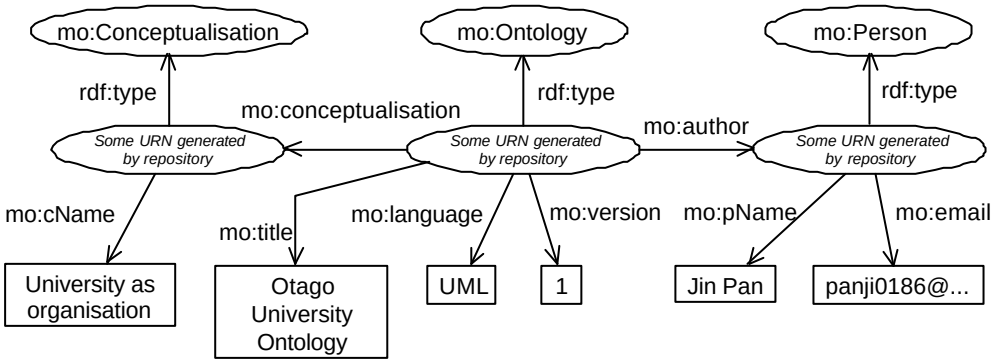


Figure 4: Sample RDF metadata about three resources: a conceptualisation, an ontology, and its author (the "mo" prefix refers to the XML namespace for the meta-ontology RDF schema)

Usage Scenarios

Figure 2 shows the process of publishing an ontology in the repository using POST. The final arrow illustrates how the returned URN can subsequently be used to record additional information about the ontology resource using Rdftp.

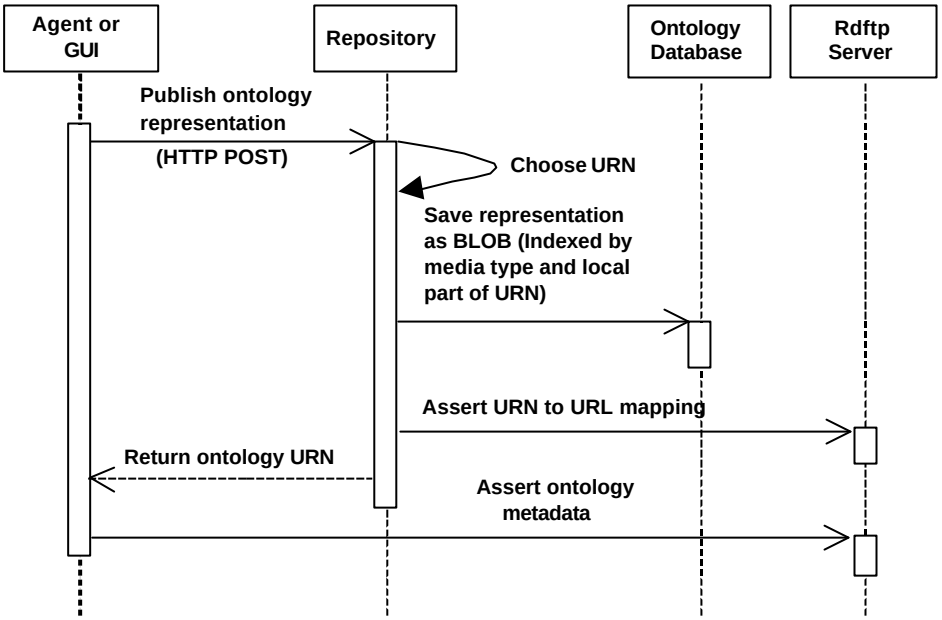


Figure 5: Publishing an ontology using HTTP POST

Figure 3 shows how an OPAL agent can query the ontology metadata to find an ontology of interest, resolve the URN and then retrieve the ontology.

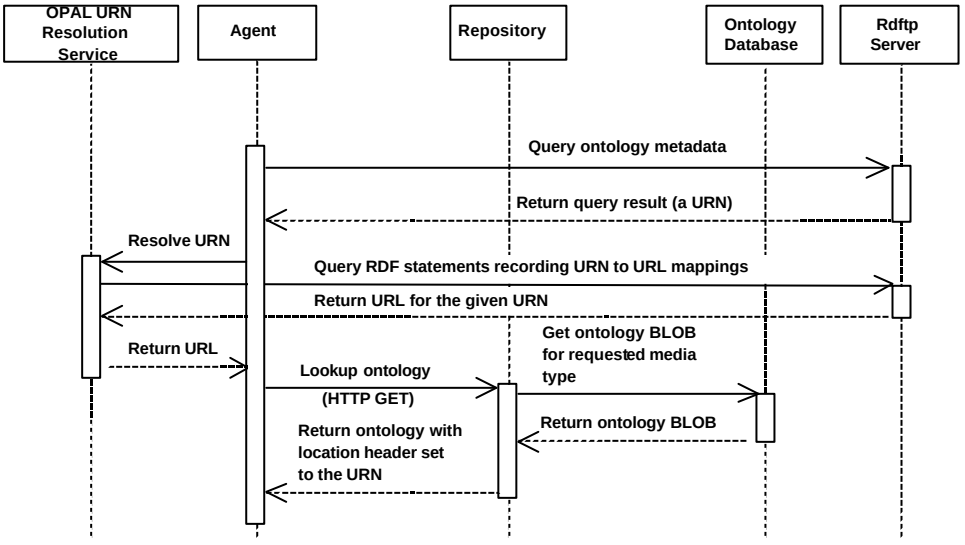


Figure 6: Locating an ontology by querying the repository

Overview of the Organisation

The University of Otago is New Zealand's oldest university. The Software Agents Research Centre in the Department of Information Science, together with colleagues in the Department of Computer Science, have established the university as New Zealand's leading centre for agent research. We are active members of FIPA and maintain the Dunedin node on the Agentcities network.

Acknowledgements

Thanks to Mannes Poel from the University of Twente in The Netherlands for demonstrating this system at Agentcities iD3. This research was funded by a grant from the University of Otago.

References

- [1] FIPA. FIPA 98 Specification Part 12: Ontology Service. http://www.fipa.org/specs/fipa00006/OC00006A.html, 1998.
- [2] Chaudhri, V. K. et al. OKBC: A Programmatic Foundation for Knowledge Base Interoperability, Proceedings of the 15th National Conference on Artificial Intelligence, 600–607, 1998.
- [3] Fielding, R. Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology, 2(2) 115–150, 2002.
- [4] Vassiliadis, V. Rdftp: RDF Server. http://semanticweb.gr/rdftp 2002

I5: WSDLTool

Tool for Deploying Web Service Wrapper Agents

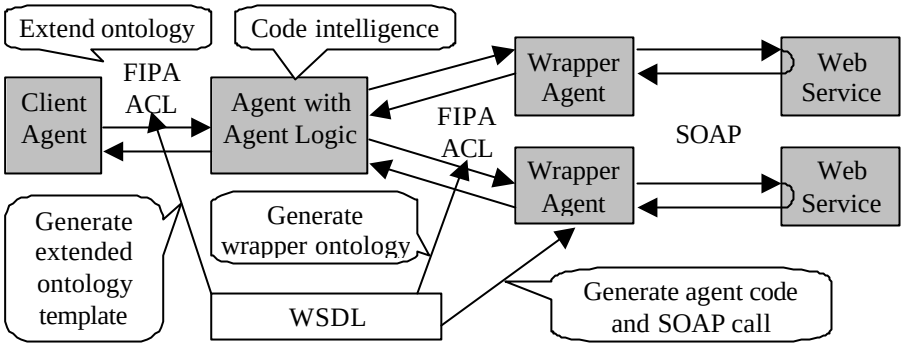
System Development Department
Computer and Automation Research Institute (SZTAKI)
Budapest, Hungary
laszlo.varga@sztaki.hu

Abstract
The WSDLTool supports the integration of existing web services into agent systems like the Agentcities network. The input of the tool is a WSDL file describing an existing web service. The tool generates code for the deployment of a wrapper agent which can accept FIPA request messages from client agents to execute a web service, invoke the web service and inform the client agent of the results of the web service. With the help of the WSDLTool the developer may generate code for agent deployment in different formats: Protégé project file, RDF Schema, JADE Ontology code in Java, Java code for agent deployment on JADE platforms. WSDLTool helps to bridge the gap between the theoretical foundations of agent technologies and their potential for industry-wide deployment.

Brief Description

Because the agent oriented approach is considered as one of the most important models for engineering complex distributed systems, research on methods for agent engineering became important. Methods for engineering agent systems from scratch are well developed and methods for creating agent systems from existing applications are needed as well. In most of the cases agent systems are developed on top of a legacy system where agents have to be able to access the legacy system through a non-agent protocol and they have to be able to communicate with each other using an agent language. By developing methodologies for integrating existing systems into agent systems we help to bridge the gap between the theoretical foundations of agent technologies and their potential for industry-wide deployment. Nowadays there are standards for these interfaces: agent communication is standardised by FIPA and the web service interface characterised by XML, SOAP, UDDI and WSDL is becoming widely used for accessing Internet applications, therefore it is promising to integrate existing systems into agent based systems using the FIPA and the web services standards.

Since existing web services and agent systems use different protocols in their communications (namely web services typically use SOAP protocol over HTTP and agents use ACL messages over IIOP) client agents cannot access the web services directly. For this reason some kind of proxy solution is needed to convert agent requests into the appropriate web service calls and web service results into agent communication language, as shown on the Figure below. The grey boxes show the components of the agent system based on web services and the white boxes show what the agent system developer can do with the help of the WSDLTool. From the WSDL description of the web services the developer can generate the code of the wrapper agent, the ontology for the communication between the wrapper agent and its client agents (typically an intelligent agent that combines the web service results and adds autonomy as well as proactive behaviour), the ontology template for the communication between the intelligent agent and its clients. The code of the intelligent agent and the ontology for its clients has to be extended manually to include agent logic. The WSDLTool has two parts to support the creation of agent systems based on web services: WSDL2JADE and WSDL2PROTÉGÉ.



WSDL2JADE

The goal of WSDL2JADE is to process the web service description file (WSDL) as the only input and automatically generate agent ontology and agent deployment code implementing the wrapper agent of the web service in Jade platform.

The generated agent code can be divided into two parts: ontology related code and agent code. The ontology code defines the entities and the semantics of the information which can be passed between the client agents and the wrapper agent, namely the web service operations, the input parameters needed by the web service call and the return values of the service call. The agent code implements the wrapper agent which waits for client agent requests, interprets the request parameters, converts them into the proper web service protocol, calls the web service, and converts the results back to agent communication language.

Due to the complexity of the WSDL structure, the WSDLTool first builds an internal WSDL model extracting the relevant information from the WSDL file. In the next step the ontology and the agent code are generated from the internal model. The web service data types defined in the WSDL are mapped to ontology data types. The web service invocation is mapped to an ACL message of FIPA REQUEST type corresponding to an agent action. The web service result return is mapped to an ACL message of FIPA INFORM type corresponding to a predicate in the ontology. Once the agent code has been generated it can be deployed and started on a Jade agent platform.

WSDL2PROTÉGÉ

WSDL2PROTÉGÉ is similar to WSDL2JADE, but it generates a Protégé project file as output. The output Protégé project file can be loaded into Protégé knowledge engineering tool and can be used to engineer the wrapper agent in two different ways.

In the simplest case the wrapper agent is engineered without modifying the generated Protégé project file. In this case the agent engineer can generate the Jade ontology code of the wrapper agent from Protégé and has to write the logic of the wrapper agent by hand. This is not the recommended usage of WSDL2PROTÉGÉ, because this is automatically done by WSDL2JADE. The only added value of WSDL2PROTÉGÉ in this case is that the RDF Schema and the Html documentation of the wrapper agent ontology can be generated from Protégé.

In a more sophisticated case the WSDL2PROTÉGÉ is used to generate the stub of the ontology for the intelligent agent and the extended ontology as well as the intelligent agent is engineered by modifying the generated Protégé project file. In this case the agent engineer can define additional concepts, actions and predicates in the generated Protégé project file using Protégé and then the agent engineer can generate the Jade ontology code of the extended wrapper agent

from Protégé and can write the logic of the extended wrapper agent by hand. Since the added intelligence cannot be generated from the WSDL file, the extended wrapper agent code cannot be engineered with WSDL2JADE, but with the WSDL2PROTÉGÉ tool.

Usage

WSDLTool can be accessed from: http://sas.ilab.sztaki.hu/wsdltool/.
The web interface provides access both to WSDL2PROTÉGÉ and WSDL2JADE. The user first has to specify the location of the WSDL file which can be either a local file or a URL. Next, the user has to specify the type of the output which can be either "Protégé project files" or "Jade agent code". In the first case when clicking on the "Generate the zip file..." button the processing servlet will create project files for Protégé according to the input WSDL, and packs them into a zip file. In the second case Jade agent code will be packed into the result zip file. The result zip file can be saved on the local disk. After unpacking the zip file the user can load the files into Protégé or launch the agent in Jade platform.

Usage scenario

In order to test the tool use for example the NumberSpeller web service (its correct WSDL file can be found on the a starting page of the WSDLTool) from the following web page:
http://dwdemos.alphaworks.ibm.com/framework/abpage.html

In order to test the WSDL2PROTÉGÉ tool generate a Protégé project file, download the result and load it into Protégé where you can browse and edit it. Generate the RDF Schema, the html documentation and the Jade code of the ontology from Protégé as described in the Protégé documentation. Look at the generated Jade agent code, the Jade manual and a web service tutorial to see how you could develop a complete wrapper agent if you had time.

In order to test the WSDL2JADE tool generate Jade agent code and download the result. Deploy the generated ontology and agent code in Jade as described in the Jade documentation. Develop a client agent or use the dummy agent of Jade to test the generated wrapper agent.

An application demonstrating the output of the WSDLTool is available from http://sas.ilab.sztaki.hu/wsid/

I6: X-Security

Communication Security in Multi-Agent Systems

Agent Technology Group
Gerstner Laboratory, Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague,
http://agents.felk.cvut.cz

Abstract
Both, research and application development in the area of multi-agent systems currently undertakes rapidly expansion. In order to use multi-agent technology in real applications, it is inevitable to ensure security, integrity and authenticity of inter-agent communication. The security systems, developed for different applications are used for this purpose. Alternatively, multi-agent systems (MAS) are designed with respect to the specific communication security requirements. This paper describes a universal security system, which implements authentication and secure communication among agents. System ensures full cooperation of agents using the security and already existing (unsecured) agents. Principle of the system integration into an already existing MAS is described here as well as its usage for mobile agents, security system's activity during inaccessibility of certification authority (CA) and possibility of CA's recreation. Security system was designed according FIPA standards.

System Description

Assuring a safe and secure communication for inter-agent communication is a must when designing and implementing industrial or e-business applications. We have designed and implemented the **X-Security** package for secure communication as a part of AgentExchange, the **Agentcities.NET** project. This package is implemented in **Java** as an extension of **JADE** multi-agent platform. Instead of using Java Secure Socket Layer (SSL), we have decided to extend the FIPA ACL message structure with a new **X-Security** slot. In this slot we specify, how the message content has been signed/encrypted, what certificate (pair of keys) was used, etc. This approach allows portability between different types of platforms. Using this mechanism, chaining and storing of signed messages are also possible, unlike in SSL where only one-time communication channels are opened. Either the whole message content or just parts can be signed/encrypted. Agents can exchange both secure and unsecured messages at the same time. This means, that agent not using a security package can use current form of ACL Message without any restrictions.

Our security package offers these two basic functions:

- **Security Certification Authority (SCA)** - stand-alone agent, application independent, which allows certificate registration/deregistration and provides other agents in community with these certificates. It isn't a part of the platform (like AMS or DF).
- **Security module** - this module has to be included into an agent that wants to use a communication security. Structure of the agent with integrated security module is shown on Figure 1.

Security package currently uses these algorithms for communication security:

- **Message signing** - SHA with DSA algorithm

- **Message encrypting** - RSA algorithm (using BouncyCastle security package)

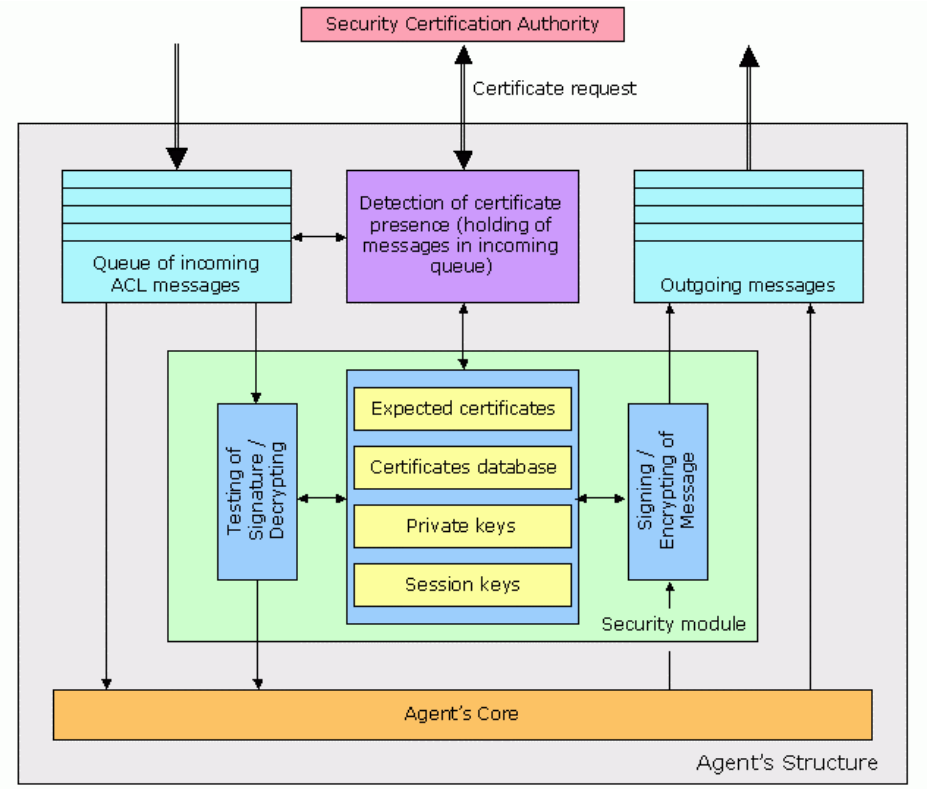


Fig. 1: Agent's structure with integrated security module

System Usage

Our security model was developed as an add-on for JADE multi-agent platform. This model was implemented as a library allowing easy connection to platform without necessity to edit its source codes. Only minor changes to the source codes of already existing multi-agent systems are required when including this security package. Security package is public available and can be downloaded at:

http://agents.felk.cvut.cz/security

Security Certification Authority is currently running on Agentcities node in Prague. Other agents can register or deregister their own security certificates (generated using the X-Security package) here. Agent can also search its certificate database for requested certificate (by agent's name or certificate identifier).

Organization Description - Agent Technology Group

Mission statement:

Currently, agent modelling, inter-agent communication, software integration, social intelligence, coalition formation, community monitoring and meta-reasoning issues are investigated in the group. Applied research focuses on design and development of multi-agent systems and on improving their collaborative behaviour. Three baseline models were developed in the group - ProPlanT (Production Planning Tool - agent based production planning system), eXPlanTech (FIPA compliant system for production planning and supply chain management) and CPlanT (Multi-agent System for Humanitarian Relief Operation Planning).

Exhibitors

Stand	***Reception***	***Competition and Exhibition Reception***
S1 & 2	*Emma Rollón, Carlos Mérida-Campos, Andrey Girenko*	Reception, registration and general event information. Come here to: • Registration • Get details of the iD3 Social events, lunch and coffee • Get directions to anywhere on campus or in town • Sign Agentcities.NET Travel claim forms

Stand	***Prizes***	***Prizes and Promotional Materials***
S3	*Emma Rollón, Carlos Mérida-Campos*	Information and promotional materials from Sponsors including: • Prize Gallery • Competition Information • Call for Papers on Related Conferences and other informational materials

Stand	***Consorts***	***Cognitive Resource Managing Agents in Ubiquitous Computing***
A1	*Cyber Assist Research Center, AIST*	CONSORTS (Architecture for COgNitive reSOurce management with physically-gRounding agents) is a new kind of agent architecture for Ubiquitous computing, which is designed to realize mass-user support in addition to conventional personal assistance. The key concepts in Consorts are "physically-grounding" and "cognitive resources". By using sensory information brought by ubiquitous environment, agents have grounding to physical world and they are conscious of physical resources (especially spatio-temporal resources) in a cognitive way, i.e., they can recognize, reorganize, and operate raw physical resources as cognitive resources. Services realized in Consorts include 1) extension of conventional personal services using physical information such as position, and 2) mass-user support that realizes information providing and social coordination for mass-user as mass beyond personal support.

Stand	***GruSMA1***	***Deployment of agent-based health-care services***
A2	*University Rovira i Virgili*	In this project we have implemented and deployed an agent-based AgentCities platform that offers medical services to the citizens and visitors of a city. These services allow a user to a) make queries related to information about medical centres, medical departments and doctors, b) access his/her medical record and c) make an appointment to be examined by a particular doctor. At the examination time, the doctor is able to access and update the patient's medical record. Special attention has been paid to the definition of a medical ontology in RDF and to the implementation of mechanisms that guarantee the confidentiality of medical information.

Stand	***Ibundler***	***a decision support service for negotiating agents***
A3	*"Los Chicos", iSOCO, S.A.* Multi-item negotiations in industrial procurement pose serious challenges to buying agents when trying to determine the best set of selling agents' offers. Typically, a buying agent's decision involves a large variety of constraints that may involve attributes of separate items as well as attributes of different, multiple items. IBundler is intended as a negotiation service offered to buying agents to help them determine the optimal bundle of received offers based on their constraints and preferences. In this way, buying agents are relieved with the burden of solving too hard a problem (the problem can be proven to be NP[4]) and concentrate on strategic issues. IBundler is intended as a negotiation service for buying agents and as a winner determination service for combinatorial auctions. To the best of our knowledge this is the first agent service for such type of complex negotiation scenarios.	

Stand	***IRES***	***On the Integration of Restaurant Services***
A4	*Agents Research Laboratory - Universitat de Girona* Our contribution to AgentCities is a restaurant recommender service that consists of a multi-agent system of service agents and personal agents. The service agents offer information about restaurants and the personal agents. The personal agents provide restaurant recommendations to their users based on information on restaurants and interaction with other *friendly* personal agents. To achieve this purpose, the personal agents interact with the restaurant server agent to find information about the restaurants, with the personal agent facilitator agent to find out other personal agents in the system, and with other personal agents to find similar users and benefit from their opinions and advice. In order to improve the performance of the personal agents, case-based reasoning and trust techniques have been applied.	

Stand	***MAS-VPR***	***MAS Vehicle Routing Problem with Time Windows***
A5	*St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science* The entry is auction-based multi agent system for Vehicle Routing Problem with Time Windows (VRPTW). The application was implemented to solve a well-known benchmark developed by Solomon. This benchmark incorporates a shipping company possessing a depot and a number of trucks and a set of orders from different customers. Given constraints the task is to allocate orders over trucks. At that number of trucks and total distance covered by them should be minimal. The system to be demonstrated includes *Shipping company* agent playing the role of auctioneer and several *Truck* agents playing the roles of contractors. Specification of the application developed is represented in a specific language that is supposed by the MAS development tool used. This tool also used for code generation and deployment the application in a local network. Applied MAS is provided for user interface that allows visualizing results of auctions and supports interactive mode of task solving.	

[4] Actually the problem faced by iBundler is more general than the winner determination problem in combinatorial auctions.

Stand	***Settler***	***Agent-based Settler Game***
A6	*University of Hamburg, Faculty of Informatics* In the context of agent oriented software development we use a settler game to illustrate our approach to architecture, modelling, and implementation. The approach is aiming at open agent systems. The development is based on a distributed and truly concurrent framework including Reference nets and Mulan illustrated below. Settler itself is a multi-player game with a lot of possible interactions between agents.	

Stand	***TAGA***	***An Travel Market Framework in Agentcities***
A7	*UMBC-TAC, U. Maryland and Baltimore County* Travel Agent Game in Agentcities (TAGA) is a framework and infrastructure that extends and enhances the original Trading Agent Competition (TAC) system [Wellman 99] to work in the Agentcities environment. Auction service is added to enrich the Agentcities service feature. RDF and WSDL are used to enhance the interoperability of FIPA compliant agent and web service. The power of Agentcities technology plays a big role in building this distributed and open market framework, which further facilitates the web-based e-commerce research.	

Stand	***EFA***	***Embedded FIPA Agents***
I1	*Media Lab Europe* This project aims to deploy agents on a self-contained, FIPA-compliant agent platform hosted by an embedded device. The aim is to allow agents to be directly embedded into physical spaces and objects in a manner not supported by the typical beige-box PCs. *Fraggle Rock is the smallest Agentcities platform currently deployed, measuring just 3x2.5" and running a modified version of the JADE/JADE-LEAP platforms. It remains FIPA-compliant and includes a number of innovations important to open, dynamic environments.*	

Stand	***Entish***	***eLingua for service description and composition***
I2	enTish Team, Polish Academy of Sciences A new technology for service description and composition in open and distributed environment is proposed. The technology consists of description language (called Entish) and composition protocol called entish 1.0. They are based on software agent paradigm. The language and the protocol are merely specifications. To prove that the technology does work, the prototype implementation is provided. It is still under testing and is available for use and evaluation via www.ipipan.waw.pl/mas/. The specifications were created on the basis of the requirements produced by the Service Description and Composition Working Group within the framework of Agentcities.NET. Related work was done by WSDL+BPEL4WS+(WS-Coordination)+(WS-Transactions), and DAML-S.	

Stand	***ORCAS***	***Open, Reusable and Configurable multi-Agent Systems***
I3	IIIA-CSIC/IBROW	
	ORCAS is an electronic institution designed to configure multi-agent systems on-demand, by using a knowledge modelling approach to describe agent capabilities. Specifically, we propose Problem-Solving Methods as the paradigm to describe and reason about agent capabilities. The configuration of a MAS is performed in two steps: capability-brokering, which goal is to find a configuration of agent capabilities satisfying the requirements of a problem, and team-formation, that uses the configuration of agent capabilities to form a team of agents instructed to solve that problem in a cooperative way.	

Stand	***Repository***	***The Otago Ontology Repository***
I4	*University of Otago, Dunedin, New Zealand*	
	The Otago Ontology Repository is a prototype implementation of a design to allow ontology designers and agents to use open Web standards to publish and retrieve ontologies and metadata about them. Key features of the system include the use of the HTTP protocol following the REST architectural style, the representation of the recorded information about ontologies and the repository information schema using RDF and its schema language RDFS, and the use of URNs to identify ontologies.	

Stand	***WSDLTool***	***Tool for Deploying Web Service Wrapper Agents***
I5	*Computer and Automation Research Institute (SZTAKI)*	
	The WSDLTool supports the integration of existing web services into agent systems like the Agentcities network. The input of the tool is a WSDL file describing an existing web service. The tool generates code for the deployment of a wrapper agent which can accept FIPA request messages from client agents to execute a web service, invoke the web service and inform the client agent of the results of the web service. With the help of the WSDLTool the developer may generate code for agent deployment in different formats: Protégé project file, RDF Schema, JADE Ontology code in Java, Java code for agent deployment on JADE platforms. WSDLTool helps to bridge the gap between the theoretical foundations of agent technologies and their potential for industry-wide deployment.	

Stand	***X-Security***	***Communication Security in Multi-Agent Systems***
I6	*Gerstner Lab, Czech Technical University in Prague*	Both, research and application development in the area of multi-agent systems currently undertakes rapidly expansion. In order to use multi-agent technology in real applications, it is inevitable to ensure security, integrity and authenticity of inter-agent communication. The security systems, developed for different applications are used for this purpose. Alternatively, multi-agent systems (MAS) are designed with respect to the specific communication security requirements. This paper describes a universal security system, which implements authentication and secure communication among agents. System ensures full cooperation of agents using the security and already existing (unsecured) agents. Principle of the system integration into an already existing MAS is described here as well as its usage for mobile agents, security system's activity during inaccessibility of certification authority (CA) and possibility of CA's recreation. Security system was designed according FIPA standards.

Stand	***LegalAdvisor***	***Acklin B.V.***
S1	*Ruurd Pels and Chris van Aart*	Acklin will show the "LegalAdvisor" system which was developed for the AgentCities project. It shows collaboration between different types of agents to determine applicable law. Also, Acklin shows web based visualisation of agent interaction and web based creation of crosstab graphs on agent messages collected in a database.

Stand	***Morpheus***	***Online reservations in the Maastricht.agentcities.net project***
S2	*Gabriel Hopmans and Peter Kruijsen*	For the Maastricht.agentcities.net project the University Maastricht, Morpheus Software and other collaborators developed an online reservation system. In the demonstration we will show the system so far, we will show how restaurant managers can edit their restaurant information and how people can make a choice for their restaurant and menu. We will illustrate how agents can provide support. For the presentation we will also discuss the problems we encountered and which problems still have to be solved. As last point we will talk about our ideas how the system will be further developed and how it can be commercialised for the future.

Stand	***INCA***	***Sheffield Hallam University***
S3	*Martin Beer*	Sheffield Hallam University will demonstrate a prototype Community Care System developed using an Agentcities.NET deployment grant.

Stand	***Whitestein***	***Whitestein Technologies***
S4	*Monique Calisti, Martin Kernland*	
	Whitestein Technologies was founded in 1999 with the mission to become a leading provider of advanced software agent technologies, products, solutions, and services for various applications in several industrial domains. We strongly believe that agent-based technologies will be the foundation of a next generation of distributed information systems and network infrastructures, in particular in combination with other leading-edge technologies such us web services and mobile wireless computing. Based in Switzerland, Whitestein Technologies as of today has a team of about 50 enthusiastic and highly skilled people, with offices in Zurich (Switzerland), Bratislava (Slovakia), and Sophia Antipolis (France). Whitestein Technologies has a strong background and experience base in the industrial and commercial IT-world, so we are particularly aware that software agent technologies will only thrive in the marketplace if they reach industry-grade quality and applicability, and can be well integrated with "traditional" technologies and products.	

Stand	***Magenta***	***Magenta Supply Chain and Logistics Solutions***
S5	*George Rzevski*	
	MagentA Corporation markets high granularity multi-agent systems for Real-Time Logistics, eCommerce and Knowledge Management. The demonstration will include a production scheduling prototype developed for a major European car manufacturer, a logistics program for CargoLifter Network and tools for the rapid construction of ontologies.	

Stand	***iSOCO***	***Intelligent Software Components, S.A.***
S6	*Richard Benjamins*	
	iSOCO - Intelligent Software Components, S.A. is a company founded in 1999 by a group of researchers from the Artificial Intelligence Research Institute (IIIA) of the Spanish Scientific Research Council (CSIC) in Barcelona. Currently, iSOCO employs 90 persons distributed over offices in Barcelona, Madrid and Valencia. iSOCO's mission is to help clients get more value out of the information they manage by devising intelligent software for the networked economy. Apart from offering customized, innovative business solutions to clients, iSOCO is active in R&D at an international level (e.g. four European IST projects), and commercializes three products: GETsee: a personal content aggregator for financial and other online information; Quotes: a component framework to setup complex and specific corporate sourcing solutions; and Vis@do: a tool to securily digitize workflow in large organizations. iSOCO has provided innovative solutions to more than 60 clients, and has partnerships with more than 20 organizations. More information at www.isoco.com.	

Stand	***Granite Nights***	***University of Aberdeen***
S7	*Gunnar AAstrand Grimnes and Alun Preece* *An Agent-Based Evening Scheduler for the Granite City* Granite Nights is a multi-agent application which allows a user to schedule an evening out in Aberdeen (aka the Granite City). Agents within the application infrastructure are organised according to a series of pre-defined roles. These are: information agents (wrappers for RDF resources - either static or dynamically generated from existing WWW sources); profile agent (manages user data, such as id, password and preferences); constraint-solver agent (maps RDF data to finite domain constraints and produces valid instantiated schedules); evening-agent (receives user queries and controls invocation of other agents to generate a solution); user-interface agent. The application has been constructed using the JADE agent platform, SICSTUS Prolog (+Jasper) as well as BlueJADE running under the JBOSS application server. Granite Nights knows about three different types of events/venues in Aberdeen, pubs, restaurants and films showing at local cinemas. Using the intuitive web-interface the user can guide the composition of these to generate an evening to suit him. We will demonstrate the intuitive web interface, as well as show some of the internal interaction between the Agents that make up Granite Nights.	

Stand	***UPM***	***Universidad Politécnica de Madrid***
S8	*Universidad Politécnica de Madrid (CLIP and ISYS groups)* Agent Technology activities at UPM are lead by the CLIP and ISYS groups who will demonstrate a range of systems developed for National and European level research activities they participate in.	

Stand	***UPC***	***Universitat Politècnica de Catalunya***
S9	*Universitat Politècnica de Catalunya (KEMLG, AI and LSI)* The Universitat Politècnica de Catalunya (UPC) is the local host for Agentcities ID3 and the Agent Technology Exhibition. UPC is one of the best-known Universities in Spain and makes contributions to National and European research in many domains. In the area of Agent technology UPC is a member of Agentlink II and Technical Coordinator of the European Commission funded Agentcities.NET and Agentcities.RTD projects.	

Stand	***agents@city group***	***CoMAS: Control and Management of Agents and their Services***
S10	*City University* Large-scale, open systems such as AgentCities pose great challenges to the management and control of agents and the services they offer. Regarding agent management and control, CoMAS provides a multi-agent system visualiser and an agent interaction monitor. The agent visualiser allows users to get an overview and investigate details of a large agent system. The agent interaction monitor will complement this approach by detecting termination of agent subsystems, which is important for timely feedback to the user, scheduling, and garbage collection. Regarding service management, we have developed a market-mechanism to allocate computational resources. To this end, different strategies are available depending on the nature of the requirements such as e.g. timeliness of delivery, quality, speed. This will be complemented by an intelligent trader, which composes services using an expressive language, which can capture a variety of constraints to express valid and invalid compositions. In the demo we will show the marketplace for resource allocation and the agent visualiser.	

Stand	***Agentcities.RTD***	***Advances in Network Architecture***
R1	*Agentcities.RTD* This stand will show progress on development of new, more flexible network service components to support the Agentcities network. Basic information on how to: • Connect to the Agentcities Network • Deploy a simple service Will also be available.	

Stand	***Agentcities.RTD***	***Composable Business Services – Event Planner***
R2	*Agentcities.RTD* This stand will demonstrate prototypes of the business component services being developed by the Agentcities.RTD project and show how they can interact dynamically across the Agentcities network to provide compound business services.	

Stand	***Agentcities.RTD***	***Composable Content Services – Evening Organiser***
R3	*Agentcities.RTD* This stand will show Information and content services being developed by the Agentcities.RTD project that can be used as building blocks to create a simple evening organiser application.	

Stand	***Meeting***	***Open Meeting/ Networking Space***
M1	*Unmanned*	
	Free meeting space for use on demand by event participants	

Stand	***Meeting***	***Open Meeting/ Networking Space***
M2	*Unmanned*	
	Free meeting space for use on demand by event participants	

Stand	***Meeting***	***Open Meeting/ Networking Space***
M3	*Unmanned*	
	Free meeting space for use on demand by event participants	

Stand	***Meeting***	***Open Meeting/ Networking Space***
M4	*Unmanned*	
	Free meeting space for use on demand by event participants	

Stand	***Meeting***	***Open Meeting/ Networking Space***
M5	*Unmanned*	
	Free meeting space for use on demand by event participants	

Sponsors & Supporters

Main Sponsors

Special Thanks

Special thanks go to the City of Barcelona for their very generous support of the Prize Giving ceremony and the two main prize sponsors Fujitsu-Siemens Computer and Motorola.

Ajuntament de Barcelona

Fujitsu-Siemens Computers

Motorola[5]

[5] Motorola and the stylised M logo are registered in the U.S. Patent and Trademark Office

Whitestein Technologies

Whitestein Technologies was founded in 1999 with the mission to become a leading provider of advanced software agent technologies, products, solutions, and services for various applications in several industrial domains. We strongly believe that agent-based technologies will be the foundation of a next generation of distributed information systems and network infrastructures, in particular in combination with other leading-edge technologies such us web services and mobile wireless computing.

Based in Switzerland, Whitestein Technologies as of today has a team of about 50 enthusiastic and highly skilled people, with offices in Zurich (Switzerland), Bratislava (Slovakia), and Sophia Antipolis (France).

Whitestein Technologies has a strong background and experience base in the industrial and commercial IT-world, so we are particularly aware that software agent technologies will only thrive in the marketplace if they reach industry-grade quality and applicability, and can be well integrated with "traditional" technologies and products.

Lequia, Universitat de Girona.

Laboratori d'Enginyeria Química i Ambiental (LEQUiA) is a research group from the University of Girona, which was set up ten years ago as a dynamic framework where different academic staff could combine their research areas, from different point of views, but all of them focused to environmental aspects. Based on this idea, LEQUiA has been working in close association with several research groups on Environmental Engineering, Chemical Engineering and Artificial Intelligence, mainly with the Knowledge Engineering and Machine Learning Group (UPC). LEQUiA has a wide experience in the field of wastewater treatment plant control and supervision, mainly in the application of natural computing techniques to improve management of real plants. Though we started with a classical approach, applying our previous experience in the control of lab bioreactors, soon we realised that classical control methods show some limitations when dealing with complex environmental processes. We think that it is important to build tools that solve these processes precisely and very rapidly, because of the social, political and economic impacts of the suggested solutions. Following this approach, we concluded that it is necessary to integrate an array of specific supervisory intelligent systems and numerical computations for detailed engineering, when trying to manage optimally complex environmental problems. Main results of the research work has been concerning the development of new techniques for the design of knowledge-based systems and about their application, employing new concepts in fields such as automatic knowledge-acquisition, relevance-attribute concepts and distributed systems. Our main interests are: Environmental Decision Support Systems, Industrial-process control, Real-time system modelling, and Distributed architectures for knowledge engineering.

Tryllian

Tryllian offers a commercial-strength platform for mobile agents, the Agent Development Kit. Tryllian was founded in 1998 and is headquartered in Amsterdam, The Netherlands.

The Agent Development Kit (ADK) uses FIPA-based messaging over JXTA for communication with added reliability and security. A task-based behavior framework allows Java programmers to develop advanced agent behavior with parallel and asynchronous computing aspects easily. Third party products can also be plugged in. Tryllian believes mobile agents to be at the heart of the ubiquitous computing future. Aspects like secure deployment, relocation of components and manageable co-existence of components from different parties are essential. Already decentralized execution of business logic reduces functional complexity and increases system autonomy, aspects recognized as crucial for specific scenarios.

Tryllian focuses on dynamic business process integration. Smart components (agents) are used to emphasize and enable the evolutionary aspects of enterprise integration. Keywords are dynamism, adaptability and loose coupling.

Telecom Italia Lab

Telecom Italia Lab (http://www.telecomitalialab.com) is part of the Telecom Italia Group. The mission of TILAB is the development of innovation in Information Communication Technology through new technologies and research activities carried out in the main innovative directions, such as the architecture of fixed and mobile networks, platforms and service solutions, evolution of the internet, multimedia, ICT security, research and development of integrated silicon products (System On Chip). Its strong point is its consolidated experience in the Research and Development sector in combined action with the market prospects of the entire Telecom Italia Group.

TILAB is the leader of the JADE Project (http://jade.cselt.it), an Open Source Java agent platform that complies with the latest FIPA specifications and that provides a suitable middleware and a set of graphical tools for developing, deploying, and debugging commercial multi-agent systems, able to properly scale from J2EE down to J2ME-MIDP embedded software environments.

From the know-how and the results of research, applied innovation is obtained, able to offer advanced solutions to the Telecom Italia Group and its customers by means of a highly customized proposal divided into: highly significant consultancy, technology integration solutions, added value services and products.

OntoWeb
http://www.ontoweb.org

KTWeb

KT*web* is the key initiative of the MULTIPLE project, an Accompanying Measure to the European Commission's IST programme, which provides a broad community-building and awareness-raising service for knowledge, content and media technologies, bringing together the European projects and people working in these technologies - in research, academia and business - and offering information and resources in these domains at a global level. **KT*web*** provides information and interactive community facilities on its interactive portal, **www.ktweb.org**, and organises themed workshops to:

- help projects achieve commercialisation of their results;
- help technology and market players explore R&D issues and research activities;
- bring projects and market actors together, with a focus on project exploitation.

http://www.KT*web*.org

ISOCO

iSOCO - Intelligent Software Components, S.A. is a company founded in 1999 by a group of researchers from the Artificial Intelligence Research Institute (IIIA) of the Spanish Scientific Research Council (CSIC) in Barcelona. Currently, iSOCO employs 90 persons distributed over offices in Barcelona, Madrid and Valencia. iSOCO's mission is to help clients get more value out of the information they manage by devising intelligent software for the networked economy. Apart from offering customized, innovative business solutions to clients, iSOCO is active in R&D at an international level (e.g. four European IST projects), and commercializes three products:

GETsee: a personal content aggregator for financial and other online information; Quotes: a component framework to setup complex and specific corporate sourcing solutions; and Vis@do: a tool to securily digitize workflow in large organizations. iSOCO has provided innovative solutions to more than 60 clients, and has partnerships with more than 20 organizations.

More information at www.isoco.com.

Fundació Catalana per a la Recerca

The Catalan Foundation for Research (FCR) is a private, non-profit-making institution, whose Trust includes both enterprises and public administration. The basic objective of the FCR is to promote technological and scientific development and to disseminate it among the civil society of Catalonia.

With this aim it acts in different spheres, from different perspectives and with numerous agents (Catalan universities and research centres, local and autonomous public administrations, SME's and professional associations, R & D and technology transfer actors...), but always bearing in mind that its main mission is the promotion and circulation of science and technology as a social reality.

The activities of the FCR cover a wide range of areas, such as technology transfer; data transmission networks and Internet access; supercomputing for researchers and companies; aerospace research; electronic commerce and other spheres of the information technologies.

ECCAI - European Coordinating Committee on Artificial Intelligence

ECCAI, the European Coordinating Committee for Artificial Intelligence, was established in July 1982 as a representative body for the European Artificial Intelligence community. Its aim is to promote the study, research and application of Artificial Intelligence in Europe.

Communication technologies

Communication Technologies was established on 1st October 2000 by a management buyout of Comtec Corporation. Ex Comtec Corporation, founded in 1985, had been known in Japanese IT industry as a brave and bold `early adopter' of new technologies such as Unix, RDB, client-server model, distributed objects, internet and Java in mission-critical enterprise application systems. (Well, these technologies were at one time considered useless for real business applications!) In 1996, Comtec aimed at intelligent multi-agent as a next challenge in information systems and joined FIPA. Later, Communication Technologies participated in Agentcities, Java Agent Services and related projects. The company is headquartered in Sendai, the capital city of northeastern region of Japan where 22nd FIPA meeting was hosted in July 2001. Four sales persons are based in Tokyo branch whereas thirty developers work at Sendai HQ.

BT Exact Technologies

Agentlink II "Europe's Network of Excellence for Agent Based Computing"

AgentLink is Europe's IST-funded Network of Excellence for agent-based computing. As such, AgentLink coordinates research and development activities in the area of agent-based computer systems on the behalf of the European Commission. AgentLink supports a range of activities aimed at raising the profile, quality, and industrial relevance of agent systems research and development in Europe.

Agents Inspired Technologies S.A.

Asociación Española de Inteligencia Artificial (AEPIA)

Associació Catalana d'Intel.ligència Artificial (ACIA)

The Associació Catalana d'Intel.ligència Artificial (ACIA) is a non profit association for the advancement of Artificial Intelligence (AI) in the Catalan society. The main goal of ACIA is to support communication between people and organizations involved in AI and promote social, cultural, scientific, economic and governmental awareness on AI. ACIA is member of ECCAI (European Coordinating Committee for Artificial Intelligence), a European organization whose 27 current members are national AI associations from 26 different countries. For additional information see: http://www.acia.org/

ATC'04: Call for Interest and Suggestions

Given the strong interest in the first edition of the Agent Technology Competition and willingness to contribute on the part of many we believe that there is great value in such an event: catalysing the community and showcasing the latest technology advances.

To this end we are committed to supporting the organisation of a similar event in 2004 to keep the momentum going. A new 2004 competition might be organised by one or more projects, international organisations (FIPA, the AAMAS conference, …) or simply a group of motivated individuals and we aim to learn from this years experience to provide the new organisational team with as much support as possible.

For this reason we are seeking input and suggestions on the following topics:

- Comments and feedback on this year's organisation.
- Suggestions and ideas for a future edition: format, topics, location or anything else.
- Your provisional interest in participating in, sponsoring or helping organise a future competition.

Timetable for consultation/discussion:

- *Until 15th March '03*: Feedback, interest and ideas to competition@agentcities.org or to any of the organisers directly.
- *Until 15th April '03:* discussion on possibilities.
- *Mid May '03*: preliminary announcement of competition dates and format.

[i] JADE: http://jade.cselt.it
[ii] LEAP: http://leap.crm-paris.com
[iii] Ontology Bean Generator for JADE: http://www.swi.psy.uva.nl/usr/aart/beangenerator/

Event Sponsors & Supporters