Argumentation-based Agent Interaction in an Ambient Intelligence Context

Pavlos Moraitis[a]   Nikolaos Spanoudakis [a, b]

[a]*Dept. of Mathematics and Computer Science, Paris Descartes University*
45 rue des Saints-Pères, 75270 Paris Cedex 06, France
pavlos@math-info.univ-paris5.fr
[b]*Dept. of Sciences, Technical University of Crete*
University Campus, 73100, Kounoupidiana, Greece
nikos@science.tuc.gr

Abstract

This paper presents an argumentation based interaction within a multi-agent system designed for addressing the special requirements of accessibility content and services provisioning in an ambient intelligence context. We describe an architecture with multiple intelligent agents on the user's nomad device that address the ambient intelligence issue and cooperate with a family of dedicated personal assistance agents for each type of impairment, who are located on the server side. These agents provide the best consistent solution for people that have a combination of impairments through an argumentation based distributed decision making process. In this paper, we focus on presenting how we applied a particular argumentation framework for modelling communication and distributed decision making issues among a coalition of intelligent experts. The latter are looking for a compromise on the asked infomobility services, integrating different, very often conflicting, points of view due to the simultaneous existance of several impairments.

1 Introduction

Agent technology has been applied to the infomobility services sector in recent years. Such services include location-based services like mapping and points of interest search, travel planning and, recently, trip progress monitoring and pushing information and events to the user. Works like the ones carried out in CRUMPET (Poslad et al., 2001) and the most recent Im@gine IT (Moraitis et al., 2005) projects provided such services. The ASK-IT IP[1] furthered the challenge by aiming to support users with different types of impairments and possible combinations of impairments (e.g. a person on wheel chair that is illiterate). Moreover, ASK-IT suggested the use of ambient intelligence[2] as an enabler for providing special and context based support to impaired people while on the move. In

---

[1] Integrated Projects (IP) are European Commission co-funded projects for the Information Society Technologies (www.cordis.lu). The ASK-IT (IST-2003-511298) IP provided the requirements and partial funding for the work presented in this paper (www.ask-it.org).

[2] The concept of ambient intelligence (AmI) is a vision where humans are surrounded by computing technology unobtrusively embedded in their surroundings (www.wikipedia.org)

order to address all these challenges we proposed an agent based architecture (Spanoudakis and Moraitis, 2006) integrated with the OSGi[3] middleware.

In this paper, we focus on the most important novelty of this work, namely the use of argumentation for the distributed decision making of a coalition of assistant agents, each of them being expert on a different type of impairment. Indeed, when a user suffers from a combination of impairments, the above mentioned coalition of assistant agents, are engaged in an argumentation-based dialogue for agreeing on the needs of the user. To our knowledge, it is the first time where argumentation is used in an ambient intelligence real world application as a way to find a compromise and make a collective decision through an argumentation based dialogue. Moreover, the argumentation based interaction is combined with a standardized interaction type based on the FIPA[4] interaction protocol. We found that the application of argumentation was very natural in this context because, generally speaking, argumentation (see Rahwan, Moraitis, Reed, 2005; Parsons, Maudet, Moraitis, Rahwan, 2006) can be abstractly defined as the principled interaction of different, potentially conflicting arguments, in order to obtain a consistent conclusion. Thus, we used the argumentation based communication framework proposed by Kakas, Maudet and Moraitis (2005) relying on the argumentation framework proposed by Kakas and Moraitis (2003).

The requirements that led to this approach along with the background on the used technologies are presented in Section 2. Then, in Section 3 the architecture of the conceived Multi-Agent System (MAS) is described in detail. In Section 4 we focus in the argumentation based interaction between assistant agents. We conclude in Section 5.

2 Background

2.1 Application Requirements and Challenges

ASK-IT enhances previous projects requirements (e.g. CRUMPET, Im@gine IT and TeleCARE) on infomobility services for normal, elderly or handicapped person. It specifically introduces the following characteristics:

- Personalization encapsulates the need for knowledge regarding the situation that this person is in. Therefore, a personal assistant agent must employ powerful knowledge regarding the type of impairment of the person. The accessibility features of crossroads and busses or trains must be taken into account. For an elderly or sick person even the weather conditions make a difference to the kind of service expected. Thus, agents with knowledge sufficient to serve each impairment type must be developed. Furthermore (and for the case that a person has more than one types of impairment), these agents must be able to cooperate in order to serve that person. The system must exhibit emerging behaviour, since the amount of combinations of types of impairments can be limitless. Thus, we need a distributed decision making process

---

[3] OSGi advances a process to assure interoperability of applications and services (www.osgi.org)

[4] The Foundation for Intelligent Physical Agents (FIPA) is an IEEE Computer Society standards organization that promotes agent-based technology (www.fipa.org)

among a team of agents (i.e. experts on different impairments) that makes a consistent and common decision over one particular subject (e.g. the transport mean of a person with several impairments). This decision making process must deal with the different and often conflicting points of view of the involved expert agents and propose a solution which represents a compromise among them. In order to model this distributed decision making process we decided to use argumentation, a powerful mechanism for dealing with conflicting interactions, and, precisely, to apply the argumentation framework proposed in (Kakas and Moraitis, 2003).

- The immediate ambient plays a vital role for servicing the user (e.g. domotic services, ticketing services, etc.) and all these services must be accessible by the user agent on his/her device. The agent needs to select and get the local service and then adapt it to the user's needs using relevant knowledge and the user's profile. Moreover, the agent has access to information provided on real-time by sensors on the user's body and must act whenever an abnormal situation is recorded.

An illustrative scenario for these challenges is the case of John Agentopoulos, a person that uses a wheelchair and has heart problems. When he is moving in the city he always prefers the bus because the access is easy and he can enjoy the itinerary. However, there are some limitations or special needs for his transportation relevant to his wheelchair mobility impairment and his medical problem with his heart. John is neither an expert on the mobility needs of the wheelchair user, nor is he willing to measure his blood pressure and telephone his doctor every time that he plans a trip. Thus, whenever he plans his trip he needs his personal assistant to propose the best itinerary according to his preference, his physical and medical condition. For example, when it is snowing the system promotes the metro if John's heart rate is not normal.

2.2 The Argumentation based Communication Framework

In our work we applied the communication framework proposed in (Kakas, Maudet, Moraitis, 2005), which is based on the argumentation framework proposed in (Kakas and Moraitis, 2003). The former adopted the idea proposed in (McBurney and Parsons, 2003) that we can distinguish three languages in the representation of an agent's communication theory. A language, $\mathcal{L}$, to describe the background information that the agent has about its world at any moment and the basic rules for deciding its communication moves; a language, $\mathcal{ML}$, for expressing preferences policies pertaining to its decision of these moves; and a language, $\mathcal{CL}$, which is a common communication language for all agents.

An argumentation theory is, therefore, defined as follows:

Definition 1. A **theory** **T** is a pair, T = ($\mathcal{T}$, $\mathcal{P}$). The sentences in $\mathcal{T}$ are propositional formulae, in the background monotonic logic ($\mathcal{L}$, ⊢) of the framework, defined as L←L$_1$,…,L$_n$, where L, L$_1$,…,L$_n$ are positive or explicit negative ground literals. Rules in $\mathcal{P}$ are defined in the language $\mathcal{ML}$ which is the same as $\mathcal{L}$ apart from the fact that the head L has the general form *L=h-p(rule1, rule2)* where *rule1* and *rule2* are ground functional terms that name any two rules in the theory. This *higher-priority* relation given by *h-p* is

required to be irreflexive. The derivability relation, $\vdash$, of the background logic for $\mathcal{L}$ and $\mathcal{ML}$ is given by the single inference rule of modus ponens.

For any ground atom *h-p(rule1, rule2)* its negation is denoted by *h-p(rule2, rule1)* and vice-versa.

An **argument** for a literal L in a theory ($\mathcal{T}, \mathcal{P}$) is any subset, T, of this theory that derives L, i.e. T$\vdash$L under the background logic. The subset of rules in the argument T that belong to $\mathcal{T}$ is called the *object-level* argument. Note that in general, we can separate out a part of the theory $\mathcal{T}_0 \subset \mathcal{T}$ and consider this as a non-defeasible part from which any argument rule can draw information that it might need. We will call $\mathcal{T}_0$ the **background theory**.

The notion of *attack* between arguments in a theory is based on the possible conflicts between a literal L and its negation and on the priority relation given by *h-p* in the theory.

Definition 2. Let ($\mathcal{T}, \mathcal{P}$) be a theory, T, T' $\subseteq \mathcal{T}$ and P, P' $\subseteq \mathcal{P}$ Then (T', P') **attacks** (T, P) iff there exists a literal L, $T_1 \subseteq$ T', $T_2 \subseteq$ T, $P_1 \subseteq$ P' and $P_2 \subseteq$ P s.t.:

(i) $T_1 \cup P_1 \vdash_{min}$ L and $T_2 \cup P_2 \vdash_{min} \neg$L
(ii)   ($\exists r' \in T_1 \cup P_1, r \in T_2 \cup P_2$ s.t. $T \cup P \vdash$h-p(r, r')) $\Rightarrow$
                ($\exists s' \in T_1 \cup P_1, s \in T_2 \cup P_2$ s.t. $T' \cup P' \vdash$h-p(s', s))

Here, $S \vdash_{min}$ L means that $S \vdash$ L and that no proper subset of S implies L. Also when L does not refer to h-p, $T \cup P \vdash_{min}$ L means that $T \vdash_{min}$ L.

This extended definition means that a "composite" argument (T', P') is a counter-argument to another such argument when it derives a contrary conclusion, L, and (T'$\cup$P') makes the rules of its counter proof at least as strong as the rules for the proof by the argument that is under attack. Note that the attack can occur on a contrary conclusion L=h-p(r, r') that refers to the priority between rules.

Definition 3. Let ($\mathcal{T}, \mathcal{P}$) be a theory, T$\subseteq \mathcal{T}$ and P$\subseteq \mathcal{P}$. Then (T, P) is **admissible** iff (T $\cup$ P) is consistent and for any (T', P') if (T', P') attacks (T, P) then (T, P) attacks (T', P').
Given a ground literal L then L is a **credulous** (respectively **skeptical**) **consequence** of the theory iff L holds in a (respectively every) maximal, w.r.t. set inclusion, admissible subset of the theory.

Hence when we have dynamic priorities, for an object-level argument (from $\mathcal{T}$) to be admissible it needs to take along with it priority arguments (from $\mathcal{P}$) to make itself at least "as strong" as the opposing counter-arguments. This need for priority rules can repeat itself when the initially chosen ones can themselves be attacked by opposing priority rules and again we would need to make now the priority rules themselves at least as strong as their opposing ones.

An agent's argumentation theory will be defined as a theory $(\mathcal{T}, \mathcal{P})$ which is further layered in separating $\mathcal{P}$ into two parts as follows.

Definition 4. An agent's argumentative **policy theory, T**, is a theory $T = ((\mathcal{T}, \mathcal{T}_o,) \mathcal{P}_R, \mathcal{P}_C)$ where $\mathcal{T}$ contains the **argument** rules in the form of definite Horn logic rules, $\mathcal{P}_R$ contains **priority** rules which are also definite Horn rules with head $h\text{-}p(r_1, r_2)$, s.t. $r_1, r_2 \in \mathcal{T}$ ($h\text{-}p$ standing for higher priority) and all rules in $\mathcal{P}_C$ are also priority rules with head $h\text{-}p(R_1, R_2)$ s.t. $R_1, R_2 \in \mathcal{P}_R \cup \mathcal{P}_C$. $\mathcal{T}_o$ contains auxiliary rules of the agent's background knowledge.

We, therefore, have three levels in an agent's theory. In the first level we have the rules $\mathcal{T}$ that refer directly to the subject domain of the agent. We call these the *Object-level Decision Rules* of the agent. In the other two levels we have rules that relate to the policy under which the agent uses his object-level decision rules associated to normal situations (related to a default context) and specific situations (related to specific or exceptional context). We call the rules in $\mathcal{P}_R$ and $\mathcal{P}_C$, *Default* or *Normal Context Priorities* and *Specific Context Priorities* respectively.

In order to explain how the above argumentation framework works, we present here an example where the following theory[5] $\mathcal{T}$ represents (part of) the object-level decision rules of an employee in a company[6].

r_1: give(A, Obj, A_1)$\leftarrow$requests(A_1, Obj, A)
r_2: $\neg$give(A, Obj, A_1) $\leftarrow$needs(A, Obj)

In addition, we have a theory $\mathcal{P}_R$ representing the general default behavior of the code of contact in the company relating to the roles of its employees: a request from a superior is in general stronger than an employee's own need; a request from another employee from a competitor department is in general weaker than its own need.

R_1: h-p (r_1, r_2) $\leftarrow$higher_rank(A_1, A)
R_2: h-p (r_2, r_1) $\leftarrow$competitor(A, A_1)

Between the two alternatives to satisfy a request from a superior from a competing department or not, the first is stronger when these two departments are in the specific context of working together on a common project. On the other hand, if we are in a case

---

[5] Here and below we will use Logic Programming notation where any term starting with a capital letter represents a variable. Abusing this notation we will denote the constant names of the priority rules R and C with capital letters.

[6] Non-ground rules represent their instances in a given Herbrand universe

where the employee who has an object and needs it, needs this urgently then he would prefer to keep it. Such policy is represented at the third level in $\mathcal{P}_C$.

C_1: h-p $(R_1, R_2) \leftarrow$ common_project(A, Obj, A_1)
C_2: h-p $(R_2, R_1) \leftarrow$ urgent(A, Obj)

At this point we will present the communication framework that we will use (Kakas, Maudet, Moraitis, 2005). Agents interact using *dialogue moves* or *locutions*. The shared communication language of the agents $\mathcal{CL}$, contains a set of communication performatives of the form P(X, Y, C), where:
  – P is a performative type belonging to the set *Perf*;
  – X and Y are the sender and the receiver of the performative, respectively;
  – C is the content (i.e. body) of the locution;

The set *Perf* contains a set of performative types proposed in the literature, which are well suited for argumentation based communication. More precisely, the set of performative that we will use in this work is *Perf*={request, propose, argue, agree, challenge, refuse}.

In the framework proposed in (Kakas, Maudet, Moraitis, 2005) the communication theory of the agents is divided in three parties, namely
  – the *basic component*, T_{basic}, that defines the private dialogue steps
  – the *tactical component*, $T_{tactical}$, that defines a private preference policy of tactics
  – the *attitude component*, $T_{attitude}$, that captures general (application independent) characteristics of the personal strategy of the agent type.

The $T_{basic} \cup T_{tactical}$ is called the *tactical theory* while $T_{basic} \cup T_{attitude}$ is called the *attitude theory*. In this work we use only the tactical theory.

The *basic component* contains a set $\mathcal{T}$ of *object-level* rules defining the private dialogue steps and are (for an agent X) of the form:

  – $r_{j,i}$: $p_j(X,Y,C') \leftarrow p_i(Y,X,C)$, c_{ji}

where c_{ij} are called the enabling conditions of the dialogue step from the *performative* p_i to p_j. In other words, these are the conditions under which the agent X (whose theory this is) may utter p_j upon receiving p_i from agent Y.

The background knowledge of the agent contains also the rules:
  – $\neg p_j(X,Y,C) \leftarrow p_i(X,Y,C)$, $i \neq j$
  – $\neg p_i(X,Y,C') \leftarrow p_i(X,Y,C)$, $C' \neq C$

for every p_i and p_j in *Perf* and every subject C', C, in order to express the general requirement that two different utterances are incompatible with each other. This means that any argument for one specific utterance is potentially (depending on the priority rules

in the other parts of the theory) an attack for any other different one. Hence any admissible set of arguments cannot contain rules that derive more than one utterance.

The *tactical component* defines a private preference policy that captures the "professional" tactics of the agent for how to decide amongst the alternatives enabled by the basic part of the theory. It consists of two sets $\mathcal{P}_R$, $\mathcal{P}_C$ (i.e. *default* or *normal context* and *specific context* respectively) of priority rules.

The rules in $\mathcal{P}_R$ express priorities over the dialogue step rules in the basic part. A simple pattern that one can follow in writing these rules is to consider the dialogue steps that refer to the same incoming move

- $R^i_{k\backslash j}$: h-p($r_{k,i}$, $r_{j,i}$) ← true (or NC_{kj})
- $R^i_{j\backslash k}$: h-p($r_{j,i}$, $r_{k,i}$) ← SC_{jk}

where SC_{jk} are specific conditions that are evaluated in the background knowledge base of the agent and could depend on the agent Y, the content of the incoming locution and indeed the types j and k of these alternatives locutions. The first rule expresses the default preference of responding with p_k over responding with p_j while the second rule states that under some specific conditions the preference is the other way round. More generally we could have NC_{kj} conditions in the first rule that specify the normal conditions under which the default preference applies. Using this level is then possible to discriminate between the dialogues locutions by simply specifying that the agent will usually prefer his default behaviour, unless some special conditions are specified. The later situation can capture the fact that the strategy should vary when exceptional conditions hold. More generally this would cover any tactics pertaining to the roles of agents Y, the content, C, and other relevant factors of the current situation.

When the special conditions hold, situations of dilemma (non-determinism) in the overall decision of the theory can exist. In this case we can use a set $\mathcal{P}_C$ of higher-level priority rules, in order to resolve these conflicts and give priority to rules with the special conditions. Thus we can have rules of the form:

- $C^i_{j\backslash k}$: h-p($R_{j,i}$, $R_{k,i}$)←true (or SC_{jk})

where $R_{j,i}$, $R_{k,i}$ are priority rules of the set $\mathcal{P}_R$

3 The System Architecture

This section first provides an overview of the multi-agent system architecture and then focuses in defining the argumentation based agent-interaction that is the main focus of this paper.

3.1 The MAS Architecture

The proposed MAS architecture is based on relevant agent architectures proposed by FIPA standards and the results of the Im@gine IT project. The overall architecture with

more details can be found in (Spanoudakis and Moraitis, 2006b). Here, we briefly present the different types of agents of the proposed MAS system:

- *Personal Wearable Intelligent Device Agent (PEDA).* This agent type provides personalized infomobility services to the end-user. PEDA acts as a FIPA Mini-Personal Travel Assistant for persons with impairments. He is a lightweight agent that is typically device-dependent, such as an agent operating on a PDA[7] or laptop computer, where, for instance, bandwidth and modality become special issues.

- *Ambient Intelligence Service Agent (AESA).* This agent type connects to local area services and configures the environment of the user according to his habits/needs.

- *Personal Wearable Communication Device Agent (PWDA).* This type monitors the sensors on the user's person and provides information to the end-user. In cases of emergency he contacts the broker agent in order to find a service for declaring an emergency so that help comes as soon as possible.

- *Service Provider Agents* advertise and offer services to the ASK-IT service network. They are responsible for providing a service to the network and for advertising it to the geographically closest middle agent type (broker).

- *Broker Agent.* This type has white and yellow page information about service provider agents but also about web services scattered through the World Wide Web. Service discovery and provisioning for web services is managed by another software module the Data Management Module. He also services the service requesters by invoking the services on their behalf, returning to them the results.

- *Elderly and Disabled Assistant Agent (EDA).* This type specializes in the mobility requirements and needs of any type of mobility impairment.

- *EDA Coalition Creator (EDAC).* This agent type is responsible for accepting requests aimed for the EDA and dynamically forming the coalition of different Elderly and Disabled Assistant Agents experts that will have to deliberate on the user's goal. EDAC also acts as a provider agent because he advertises his service profile to the Broker.

In Figure 1 the overall system architecture is presented in the form of a UML[8] deployment diagram. The agent software modules/entities are depicted as small people.

---

[7] Personal digital assistants (PDAs) are handheld computers that were originally designed as personal organizers. PDAs are also known as pocket computers, palmtops, or pocket PCs.

[8] The Unified Modeling Language is a standardized specification language for object modeling. (www.uml.org).
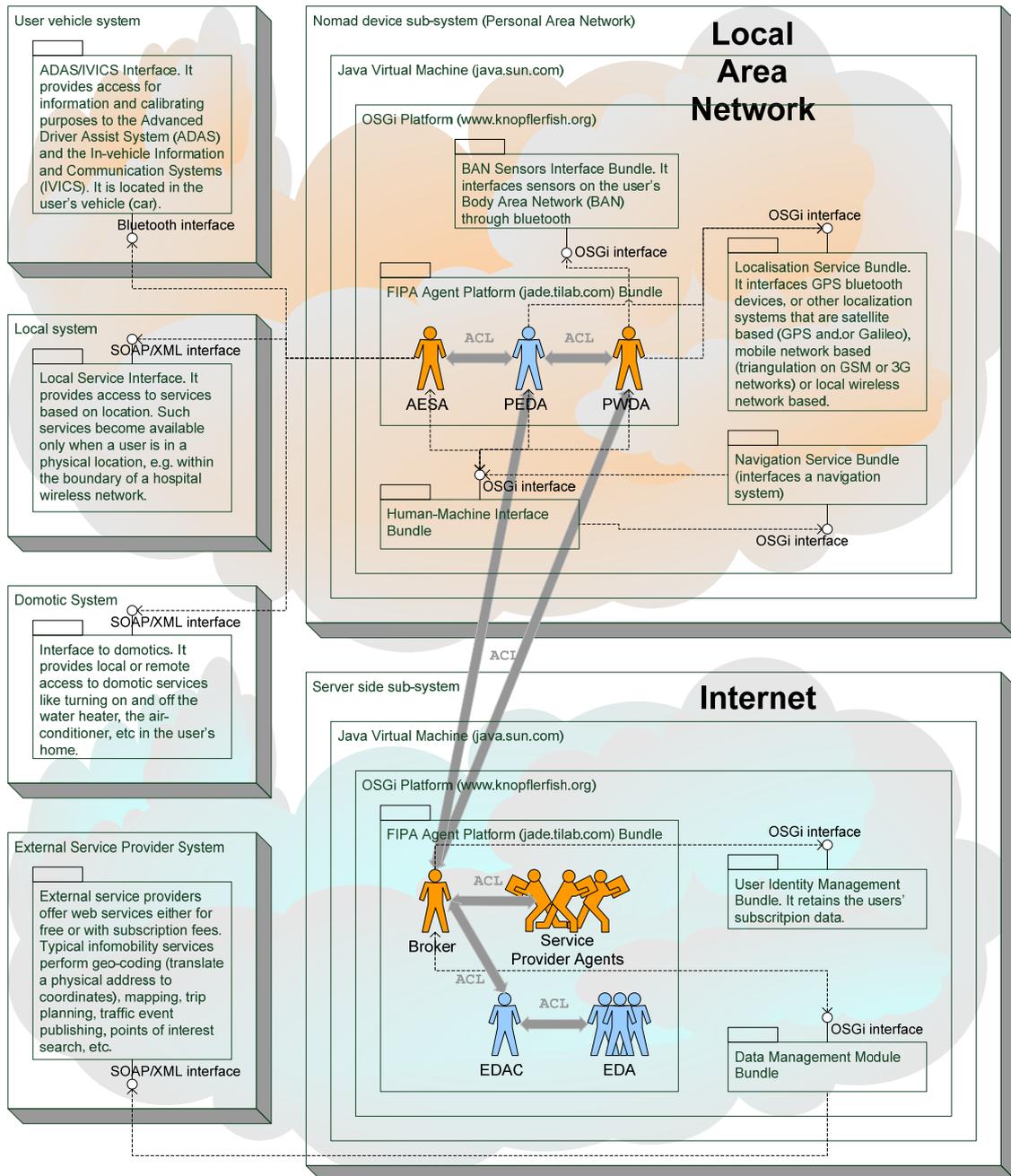
Figure 1 : The system's architecture.

The user is serviced based on his context, which relates to the following information:
- *Sensors information (monitored by the PWDA)*
  - *user heart rate*
  - *humidity*
  - *lighting conditions*
- *Local services (AESA)*
  - *indoors/outdoors, at home, in car*

- *public transport terminal LAN information*
- *current time*
- *weather report/forecast (available through web services)*

The PEDA obtains simple or subscription services from the server side using a FIPA-based *service protocol* and Agent Communication Language (ACL) that is presented in detail in Spanoudakis and Moraitis (2006a).

The main technologies involved in the development of the system presented in Figure 1 are the Knopferfish[9] OSGi framework (for integrating the MAS to the other participating software modules on both the nomad device and the server sub-systems), the JADE[10] agent platform (for developing the agents), the Jena[11] semantic web framework (for semantic matchmaking by the broker agent) and the Protégé[12] ontology editor (for developing the relevant domain ontology). For more information regarding the overall system architecture and development the reader can refer to Spanoudakis and Moraitis (2006b).

3.2   The Argumentation based Agent Interaction

In this section we present the main subject of this paper. It is considered that the knowledge regarding the needs of different mobility impaired types with regard to movement can be vast and some times conflicting if it tries to encompass all types of impairments. For this reason we decided that it will be defined separately for each type of impairment. Then, in the case of a person that has more than one type of impairments, a coalition of agents – each with the knowledge of a different type of impairment – will deliberate on the needs of the user. These *Elderly and Disabled Assistant (EDA)* agents interact through an argumentation-based communication protocol. Thus, whenever the user wants to plan a trip, the *Personal Wearable Intelligent Device (PEDA)* agent adds the context information and the user profile relevant data to the request and sends it to a broker agent that forwards it to an *EDA Coalition Creator (EDAC)* agent. The last one is responsible for finding a group of *Elderly and Disabled Assistant* agents each of them being expert in one of the user's impairments (in the case that there exist several ones). Then, the *Elderly and Disabled Assistant* agents on the server side deliberate in order to define the characteristics of the transportation mean which satisfies all the user's impairments. The result of this dialogue is sent back to the *EDAC* agent who prepares a request for the broker. The latter finds the appropriate route calculating service providers that can now service the precisely defined request (e.g. containing the desired transportation means).

---

[9] Knopflerfish project is an open source OSGi framework (www.knopflerfish.org)

[10] The Java Agent Development Environment (JADE) is an open source software framework implemented in Java language for developing multi-agent systems (jade.tilab.com).

[11] Jena is an open source Java framework for building Semantic Web applications (jena.sourceforge.net).

[12] Protégé is an open source ontology editor (protege.stanford.edu).

The FIPA interaction protocol instance (*service protocol*) presented in Figure 2 shows the sequence described above. A *PEDA* agent asks for a trip planning infomobility service and includes in his request information about the user's context. The *Broker* matches the request to his service profile repository and finds the relevant advertisement submitted earlier by the *EDAC* agent, to whom he forwards the request. The *EDAC* extracts the user impairment types from the user profile part included in the request and determines which *EDA* should be involved in the coalition that will determine the trip planning needs for this user. Then, he sends a request to these *EDA,* including the list of *EDA* participants and the original *PEDA* request. Subsequently, the argumentation dialogue takes place. It ends when all *EDA* are agreed on a proposal. The *EDA* that originally issued the agreed proposal forwards it to the *EDAC,* who integrates it in the original *PEDA* request and removes all user preference, profile and context data. Then, he sends this new request to the *Broker* who initiates a new instance of the service protocol and matches this request to the relevant trip planning *Service Provider* agent (or web service). When the *Broker* gets the results from the *Service Provider* he forwards them to the *EDAC*. Then, the *EDAC* replies to the first request of the *Broker,* so the original service protocol instance is also completed and, finally, the *Broker* forwards the results to the *PEDA*. The reader should notice that the *Broker* was involved in two instances of the *service protocol*, one initiated by the *PEDA* and served by the *EDAC* and another initiated by the *EDAC* and served by a *Service Provider*.
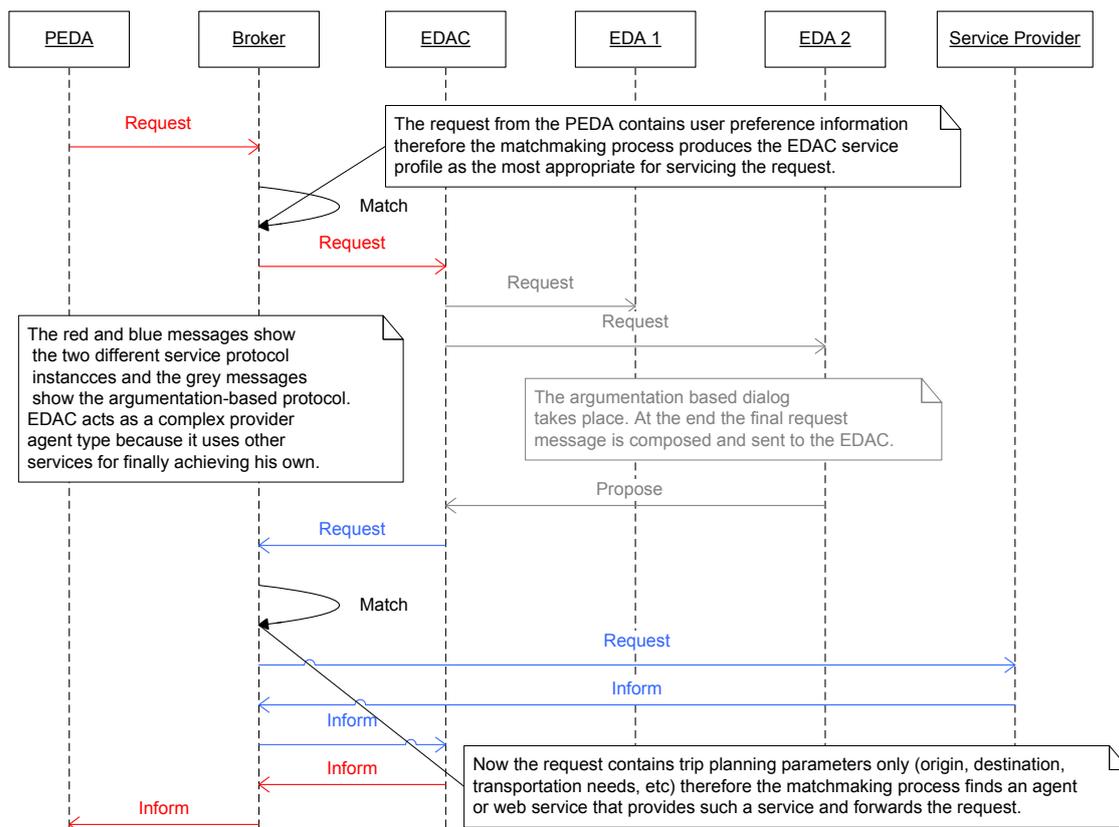
Figure 2 : The argumentation dialogue in the context of service provisioning

This argumentation based communication protocol is based on the theoretical work presented in §2.2. The strategies that are related to the agents' expertise and define their behaviour in the distributed decision making process are represented as argumentation theories. These argumentation theories are divided in two components, the first containing object level ($\mathcal{T}$) rules and the second containing default ($\mathcal{P}_R$ rules) and specific context ($\mathcal{P}_C$) rules. The first component rules represent the possible dialogue moves for the agents. The second ones represent the policies that agents can apply according to their expertise, the user profile and the different user contexts. So, in such a dialogue, the agents exchange messages whose performatives and content depend on the goal to be achieved. The possible replies to a received message are determined by its performative. The choice of the reply message type that will be sent (e.g. argue, accept, challenge), among several possible candidates, and the appropriate content, is based on the $\mathcal{P}_R$ and $\mathcal{P}_C$ priority rules, which represent the agents' preferences. This reply is a skeptical or credulous conclusion of the theory T = ($\mathcal{T}$, $\mathcal{P}_R$, $\mathcal{P}_C$) under the admissibility semantics and represents the best decision that an agent can make in a specific step of the dialogue, according to his theory (i.e. his expertise) and the information exchanged with the other agents up to that step.

We must note that in the current status of our system only bilateral dialogues are allowed. That means that if a user has more than two impairments we will have several successive bilateral dialogues. Thus, when an agreement is found between two experts the one whose proposal has been accepted by the other agent is the one who will represent both in a new bilateral dialogue with a third expert.

The argumentation-based agent interaction part of the system has been implemented using Gorgias[13] and JADE.

4. Real World Scenarios

In this section we present two real world scenarios that correspond to two use cases corresponding to the application requirements. It is obvious that for space and reader's comprehensibility reasons, we present a simplified version of these scenarios and therefore only a part of the theories that represent the knowledge of the involved agents. Thus, the only object of the dialogue is to agree for the transportation means that the user should select for his trip. In more complex scenarios the agents need to agree on more issues like the maximum number of meters that the user should walk, his preference between a fastest trip or one with fewer connections, his needs on human assistance and for services along the road (e.g. toilets), etc.

4.1 Scenario 1

A user is on a wheelchair and also has heart problems. When he is moving in the city he always prefers to use public transport (i.e. bus or metro) for economical reasons. In

---

[13] Gorgias is an open source software that implements the argumentation framework of Kakas and Moraitis, 2003 (www2.cs.ucy.ac.cy/~nkd/gorgias/)

particular, he prefers the bus because the access is easier than the metro and in addition he can enjoy the itinerary. However, due to his heart problems, some times he has to get the metro, especially when the weather is bad.

These two situations are taken into account by two different experts namely a *Wheelchair expert (WA)* and a *Heart problems expert (HPA)*. Both experts take into account the user preference for the bus (i.e. they have in their background knowledge the information that the bus is the most preferable mean), thus, only in particular situations would they propose another transportation solution, namely the metro. The WA expert proposes by default the bus as the most convenient (preferable) transportation mean. However, he can agree to the use of a different transportation mean when particular situations related to heart problems or the weather situations are presented (or both). The HPA agent needs to get the user's heart rate before advising and he also consults the weather prediction. He prefers by default the metro because the user is less exposed and therefore the risk to catch a cold, which is very dangerous for his heart problems, is minimal, but he can accept another transportation mean as well (e.g. the bus when the weather is good or when the distance to cover is short provided that it is not snowing).

These agents' behaviours can be represented through the following strategies:

- The WA Agent

-When the user asks to plan a trip he proposes by default the bus (rule r_1 below).
-When another agent proposes transportation means different to bus, he asks for explanation (rules r_2, R_1).
-However, in the case where the distance is long, he accepts the other proposal (rules r_6, R_3).
-When another agent asks him for an explanation about his own proposal to use the bus, he gives the reason, which is the fact that it is the most preferable transportation mean for the user. In normal conditions this is accepted by all the agents (rule r_2).
-When another agent proposes the bus he agrees (rule r_6, R_2).
-When he is convinced by the reasons given by another agent proposing the use of transportation means different to bus he agrees (rules r_4, R_5, R_6, C_2, C_3).
-Otherwise he sends a counter-argument, if he has an admissible one. In the specific scenario these arguments can be either the "bus is the most preferable transportation mean" (see above) or "the distance is short" (according to the situation), the later been valid even if the weather is cold (rules r_7, R_4, C_1).
-However, when the proposal concerns the taxi he always agrees (rules R_6, C_3).
-When he receives the agreement of another agent on his own proposal to use the bus he sends this proposal to the coalition creator. This declares the termination of the dialogue (rule r_5).
-When both the metro and the taxi are under consideration, it is supposed that the situation needing the taxi is mandatory and therefore he opts for the taxi (rules r_4, R_5, R_6, C_3, C_4).

The above strategies are represented by using the adopted argumentation framework through the following theories. In order to simplify the notation we will use a rule name r_1 instead of $r_{propose, \, request}$ for the object-level rules as it would be normally the case following the notation presented in §2.2. For the same reason we will use a rule name R_1 for the priorities rules instead of $R_{challenge\backslash agree}$. The same holds for all the other "r", "R" and "C" rules.

r_1: propose(X, Y, transport_mean(bus))←request(Z, X, move(Place, Place'))
r_2: challenge(X, Y, transport_mean(M))←propose(Y,X, transport_mean(M))
r_3: argue(X, Y, <R, transport_mean (bus)>)←challenge(Y, X, transport_mean (bus))
r_4: agree(X, Y, transport_mean (M)) ←argue(Y, X, <R', transport_mean (M)>)
r_5: propose(X, edac_agent, transport_mean (bus)) ←agree(Y, X, transport_mean (bus))
r_6: agree(X, Y, transport_mean(M)) ←propose(Y,X, transport_mean(M))
r_7 : argue(X, Y, <R, transport_mean (bus)>) ←argue(Y, X, <R', transport_mean (M)>)
R_1: h-p(r_2, r_6) ←transport_mean(M), M≠bus
R_2: h-p(r_6, r_2) ←transport_mean(bus)
R_3: h-p(r_6, r_2) ←transport_mean(M), M≠bus, ¬short_distance
R_4: h-p(r_7,r_4) ←short_distance
R_5: h-p(r_4,r_7) ←transport_mean(metro), ¬normal_ heart _rate(Z)
R_6: h-p(r_4,r_7) ←transport_mean(taxi), feel_ tired (Z)
C_1: h-p(R_4, R_5)←cold_weather(Day)
C_2: h-p(R_5, R_4)←snowing(Day)
C_3: h-p(R_6, R_4)←true
C_4: h-p(R_6, R_5)←true
C_5: h-p(R_4, R_5)←nice_weather(Day)

where R ∈ {most_preferable_mean(bus), short_distance} and R' ∈ {cold_weather(Day), feel_tired(Z), ¬normal_ heart _rate(Z), snowing(Day)} according to the situation

Remark 1: The WA agent has no knowledge about the weather or the user situation. So, before any interaction with the other agents who have these pieces of information the "abducibles" predicates {nice_weather(Day), ¬snowing(Day), normal_ heart _rate(Z)} (see Kakas and Moraitis, 2003) are evaluated to true. The equivalent holds for all the agents.

-The HPA Agent

-When the user asks to plan a trip he proposes the metro if the user has an abnormal heart rate (i.e. the user is less exposed in the different weather conditions namely cold or warm) (rule r_1)
-When another agent proposes the use of transportation means different to metro, he asks for explanation if this mean cannot be accepted in the current situation (rules r_2, R_1, C_6)
-Otherwise he accepts immediately (rules r_6, R_6, C_5)
-When another agent asks him for an explanation about his own proposal to use the metro, he gives the reason (rule r_3).

-When he is convinced by the reasons given by another agent proposing the use of different transportation means to the metro he agrees (rules r_4, R_4, R_5, C_1, C_3).

-When another agent proposes the metro he agrees (rules r_6, R_2).

-Otherwise he sends a counter-argument, if he has an admissible one (rules r_7, R_3, C_2). In the specific scenario these arguments can be: the "the weather is cold", "the heart rate of the user is abnormal", or "it is snowing", according to the situation. However, when the proposal concerns the taxi he always agrees (rules R_5, C_3).

-When he receives the agreement of another agent to his own proposal to use the metro, he sends the proposal to the coalition creator. This declares the termination of the dialogue (rule r_5).

-When both the metro and the taxi are under consideration it is supposed that the situation needing the taxi is mandatory and therefore he opts for the taxi considering that the taxi is not dangerous for the heart problems (rules R_3, R_5, C_3). The same holds when bus and taxi are both under consideration (rules R_4, R_5, C_4)

-We must also note that the default behaviour of this agent is to counter-argue and to accept another proposal only if specific situations hold (see rule R_3)

The above strategies are represented by the following theories:

r_1: propose(X, Y, transport_mean(metro)) ←request(Z, X, move(Place, Place')), ¬normal_ heart _rate(Z)

r_2: challenge(X, Y, transport_mean(M)) ←propose(Y,X, transport_mean(M))

r_3: argue(X, Y, < R, transport_mean(metro)>) ←challenge(Y,X, transport_mean (metro))

r_4 : agree(X, Y, transport_mean (M)) ←argue(Y, X, <R', transport_mean (M)>)

r_5: propose(X, edac_agent, transport_mean (metro)) ←agree(Y, X, transport_mean (metro))

r_6: agree(X, Y, transport_mean(M)) ←propose(Y,X, transport_mean(M))

r_7: argue(X, Y, <R, transport_mean (metro)>) ←argue(Y, X, <R', transport_mean (M)>)

R_1: h-p(r_2, r_6) ← transport_mean(M), M≠metro

R_2: h-p(r_6, r_2) ← transport_mean(metro)

R_3: h-p(r_7,r_4) ← true

R_4: h-p(r_4,r_7) ← transport_mean(bus), short_distance

R_5: h-p(r_4,r_7) ← transport_mean(taxi), feel_tired(Z)

R_6: h-p(r_6,r_2) ← transport_mean(M), most_preferable_mean(M)

R_7: h-p(r_4,r_7) ← transport_mean(M), most_preferable_mean(M)

C_1: h-p(R_4, R_3) ← ¬snowing(Day)

C_2: h-p(R_3, R_4) ← snowing(Day)

C_3: h-p(R_5, R_3) ←true

C_4: h-p(R_5, R_4) ←true

C_5: h-p(R_6, R_1) ← nice_weather(Day)

C_6: h-p(R_1, R_6) ← cold_weather(Day)

C_7: h-p(R_7, R_3) ← nice_weather(Day)

C_8: h-p(R_3, R_7) ← cold_weather(Day)

where R ∈ {cold_weather(Day), ¬normal_ heart _rate(Z), snowing(Day)} and R' ∈ {most_preferable_mean(bus), short_distance, feel_tired(Z)} according to the situation

Let's now consider a situation where a user on a wheelchair having an abnormal heart rate (this is part of the HPA agent background knowledge) wants to move from his house to his mother's house. The distance between the two places is short (this is part of the WA agent background knowledge) and the weather is cold, it is actually snowing (this is part of the HPA agent background knowledge). This situation may generate the following dialogue between an HPA and WA agent (presented graphically in Figure 3).

HPA: I propose the metro (inferred by rule r_1)
WA: I propose the bus (rule r_1)
HPA: Why the bus (rules r_2, r_6, R_1)
WA: Why the metro (rules r_2, r_6, R_1)
HPA: Because the heart rate is abnormal (rule r_3)
WA: Because the bus is the most preferable mean (rule r_3)
HPA: Yes but the weather is cold (rules r_4, r_7, R_3, R_7, C_8)
WA: But the distance is short (rules r_4, r_7, R_4, R_5, C_1)
HPA: Yes but it is snowing (rules r_4, r_7, R_3, R_4, C_2)
WA: OK. I agree (rules r_4, r_7, R_4, R_5, C_2)
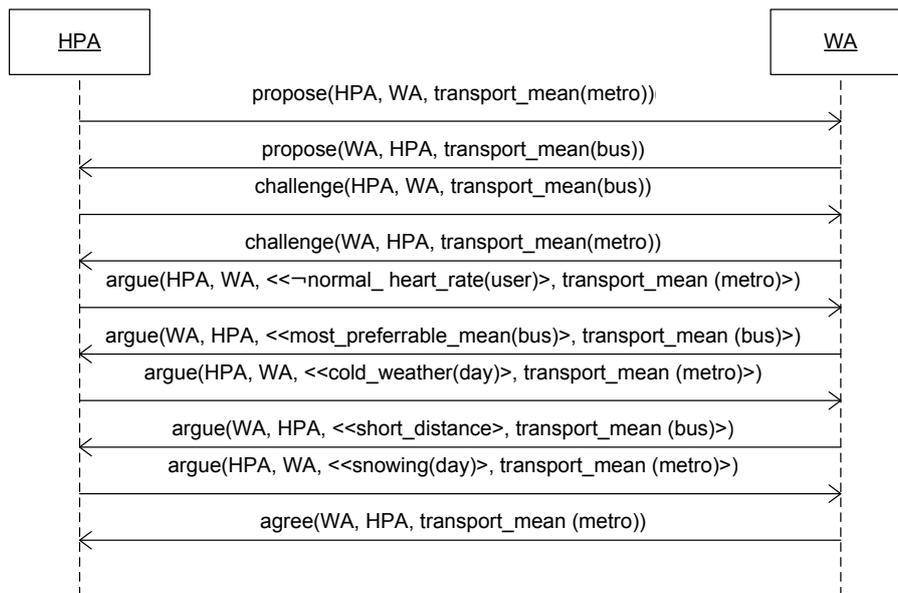
Figure 3 : The AUML sequence diagram for the 1st scenario

4.2 Scenario 2

Let's now consider a different situation where a user has heart and physical problems. In this situation a new expert, the Physical Situation agent (PSA) who cares about the user's physical situation must be involved. This agent can accept any means of locomotion provided that the user doesn't feel tired. If this is the case he cannot accept any other transportation mean than the taxi.

The **PSA agent** behaviour is presented as follows:

-When the user asks to plan a trip he makes a proposal. Particularly, he proposes the taxi when the user feels tired (rule r_1).
-In this situation this is he only solution he can accept (rules r_2, r_5, R_1).
-When another agent proposes the taxi he agrees (rules r_2, r_5, R_2, C_1)
-Otherwise (i.e. if the user is not feeling tired) he can accept the metro or the bus (rules r_2, r_5, R_3, R_4, C_2, C_3)
-When another agent asks him for an explanation about his own proposal to use the taxi he gives the reason (rule r3).
-When he receives the agreement of another agent on his own proposal to use the taxi he sends this proposal to the coalition creator. This declares the termination of the dialogue.

The above strategies are represented by the following theories:
r_1: propose(X, Y, transport_mean(taxi)) ←request(Z, X, move(Place, Place')), feel_tired
r_2: refuse(X, Y, transport_mean(M)) ←propose(Y, X, transport_mean(M))
r_3: argue(X, Y, <R, transport_mean (taxi)>) ←challenge(Y,X, transport_mean (taxi))
r_4: propose(X, edac_agent, transport_mean (taxi)) ←agree(Y, X, transport_mean (taxi))
r_5: agree(X, Y, transport_mean(M)) ←propose(Y, X, transport_mean(M))
R_1: h-p(r_2, r_5) ←true
R_2: h-p(r_5, r_2) ← transport_mean(taxi)
R_3: h-p(r_5,r_2) ← transport_mean(M), most_preferable_mean(M), ¬feel_tired
R_4: h-p(r_5,r_2) ← transport_mean(metro), ¬feel_tired
C_1: h-p(R_2, R_1) ← true
C_2: h-p(R_3, R_1) ← true
C_3: h-p(R_4, R_1) ← true

where R= {feel_tired(Z)}

Let's now consider that the user feels tired, he has an abnormal heart rate and he wants to move from his house to his girl-friend's house. This situation may generate the following dialogue between an HPA and PSA agent.

HPA: I propose the metro (inferred by rule r_1)
PSA: I propose the taxi (rule r_1)
HPA: Why the taxi (rule r_2)
PSA: The user feels tired (rule r_3)
HPA: OK. I agree (rules r_4, r_7, R_3, R_5, C_3)

5. Conclusion

In this paper we presented how argumentation can be useful in the context of ambient intelligence. More precisely, we presented a multi-agent architecture where, among different types of agents that interact through a standard communication language (i.e. FIPA-ACL) for different purposes (e.g. services providing, brokering, etc.), there exists a

coalition of assistant agents who interact using argumentation. Each of these agents is expert to a different type of impairment and they must all agree about the characteristics of a service (e.g. plan a trip) that satisfies the combination of the impairments that a user may have. By using argumentation these agents are able to solve potential conflicts and to take a consistent collective decision about the user's needs.

To our knowledge it is the first time that argumentation is used in such an ambient intelligence real world application. Through the presentation of this prototype we believe that our work proves that argumentation is very well suited for implementing complex preference reasoning mechanisms at agent individual level as well as for implementing complex interaction protocols where high level agent dialogues can take place. In the current status of our system we tested only scenarios related to trip planning needs.

Our future work is mainly concerned with modeling dialogues, where more than two agents can participate simultaneously, and with enhancing the agents' theories (and therefore the protocol) for deliberating about more types of services, not only for travelling.

Acknowledgements

References

Kakas, A., Maudet, N., Moraitis, P. "Modular Representation of Agent Interaction Rules through Argumentation", Journal of Autonomous Agents and Multi-Agent Systems, (JAAMAS), Springer, vol. 11, no. 2, pp. 189-206, 2005

Kakas A., Moraitis P., "Argumentation Based Decision Making for Autonomous Agents", Proc. 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03), pp. 883-890, Melbourne, Australia, 2003

McBurney, P., Parsons S., Argumentation based Communication between Agents. M.P. Huget (Ed.), Communication in Multi-Agent Systems, LNAI 2650, Springer-Verlag, 2003.

Moraitis, P., Petraki, E. and Spanoudakis, N., An Agent-Based System for Infomobility Services. The third European Workshop on Multi-Agent Systems (EUMAS'2005), Brussels, Belgium, December 7 - 8, 2005

Parsons S., Maudet N., Moraitis P., Rahwan I. (Eds.): Argumentation in Multi-Agent Systems, Second International Workshop, ArgMAS 2005, Utrecht, The Netherlands, July 26, 2005, Revised Selected and Invited Papers, Springer 2006

Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P. and Zipf, A., CRUMPET: Creation of User-friendly Mobile Services Personalised for Tourism. Proceedings of the second International Conference on 3G Mobile Communication Technologies, London, UK, 2001

Rahwan I., Moraitis P., Reed C. (Eds.): Argumentation in Multi-Agent Systems, First International Workshop, ArgMAS 2004, New York, NY, USA, July 19, 2004, Revised Selected and Invited Papers, Springer 2005

Spanoudakis N., Moraitis, P.: Engineering a Brokering Framework for Providing Semantic Services to Agents on Lightweight Devices. In: Proc. ECAI'06 Workshop on Context and Ontologies: Theory, Practice and Applications (C&O'06), Riva del Garda, Italy, 2006a

Spanoudakis N., Moraitis, P.: Agent Based Architecture in an Ambient Intelligence Context. In: Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS'06), Lisbon, Portugal, December 14 - 15, 2006b