Argumentation-Based Agent Interaction in an Ambient-Intelligence Context

Pavlos Moraitis, *Paris Descartes University*

Nikolaos Spanoudakis, *Paris Descartes University and Technical University of Crete*

A multiagent system uses argumentation-based interaction in an ambient-intelligence context to provide services for people with different combinations of impairments.

Recent projects such as CRUMPET (*C*reation of *U*ser-Friendly *M*obile Services *Per*sonalized for *T*ourism)[1] and Im@gine IT[2] have applied agent technology to *infomobility services*—location-based services such as mapping, points-of-interest search, and travel planning. Recently, such projects have also provided trip-progress monitoring and pushing information and events to users (that is, showing them information they didn't explicitly ask for). The ASK-IT IP (Ambient Intelligence System of Agents for Knowledge-Based and Integrated Services for Mobility-Impaired Users Integrated Project) furthered the challenge by aiming to support users having different types and combinations of impairments—for example, a person who uses a wheelchair and is illiterate. (IPs are European-Commission-cofunded projects for the Information Society Technologies research initiative; see www.cordis.lu.) Moreover, ASK-IT suggested using ambient intelligence to enable special and context-based support for impaired people on the move. To address these challenges, we proposed an agent-based architecture[3] integrated with the OSGi middleware (see www.osgi.org).

Here, we focus on ASK-IT's use of argumentation to model a distributed decision-making process for a coalition of assistant agents, each an expert on a different impairment. When a user suffers from a combination of impairments, these agents engage in an argumentation-based dialogue to agree on the user's needs. We found that applying argumentation was natural in this context because, generally speaking, we can abstractly define argumentation as the principled interaction of different, potentially conflicting arguments to obtain a consistent conclusion.[4,5] Moreover, we combined the argumentation-based interaction with a standardized interaction type based on the Foundation for Intelligent Physical Agents interaction protocol (see www.fipa.org).

Application requirements and challenges

ASK-IT extends the requirements of previous projects (such as CRUMPET and Im@gine IT) on infomobility services. It introduces several characteristics.

Personalization encapsulates the need for knowledge regarding the user's situation (including age and disabilities). So, a personal assistant agent must employ powerful knowledge regarding the user's impairment. It must also take into account the accessibility features of crossroads and buses or trains. For an elderly or sick person, even weather conditions affect the kind of service the user expects. So, we must develop agents with sufficient knowledge to serve each impairment type.

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

Furthermore, if a person has more than one impairment, these agents must be able to cooperate to serve that person. The system must exhibit emerging behavior because combinations of impairments can be limitless. So, we need a distributed decision-making process that a team of agents (that is, experts on different impairments) can use to make a consistent, common decision on one particular subject (such as the means of transportation for a person with several impairments). This process must deal with the agents' different, often conflicting viewpoints and propose a solution that represents a compromise among them. To model this process, we use argumentation, a powerful mechanism for dealing with conflicting interactions. Specifically, we apply the argumentation framework Antonis Kakas and Pavlos Moraitis proposed.[6]

The immediate ambient plays a vital role in providing user services (such as domotic or home automation services, ticketing services, and so on), and the user agent must be able to access all these services from the user's device. The agent must 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 can access information that sensors on the user's body provide in real time and must act whenever an abnormal situation is recorded.

A scenario illustrating these challenges is the case of John Agentopoulos, who uses a wheelchair and has heart problems. When he's moving in the city, he always prefers the bus because the access is easy and he can enjoy the route. However, because he uses a wheelchair and has a heart problem, he has some limitations and special needs regarding his transportation. John isn't 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 he plans a trip. So, whenever he plans a trip, his personal assistant must propose the best itinerary according to his preference and his physical and medical condition. For example, when it's snowing and John's heart rate isn't normal, the system promotes the metro to avoid the possibility of him having a serious heart situation while traveling.

The system architecture

Here we provide an overview of the multiagent system architecture and then define the argumentation-based communication framework and agent interaction that's our article's main focus.

The multiagent system architecture

Our 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 appears elsewhere.[3]

The *personal wearable intelligent device agent* (PEDA) provides personalized infomobility services to the end user. PEDA acts as a FIPA mini-personal travel assistant for persons with impairments. It's a lightweight agent that's typically device dependent, such as an agent operating on a PDA or laptop computer, where, for instance, bandwidth and modality become special issues.

The *ambient intelligence service agent* (AESA) connects to local area services and configures the user's environment according to his or her habits and needs.

> We need a distributed decision-making process that a team of agents can use to make a consistent, common decision on one particular subject.

The *personal wearable communication device agent* (PWDA) monitors the sensors on the user's person and provides information to him or her. In an emergency, the PWDA contacts the broker agent 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 provide a service to the network and advertise it to the geographically closest middle agent type (broker).

The *broker agent* has white- and yellow-pages information about service provider agents and about Web services scattered throughout the World Wide Web (that is, it has both contact information for them and information about which services each agent offers). Another software module, the data management module, manages service discovery and provisioning for Web services. The broker agent also services the service requesters by invoking the services on their behalf, returning the results to them.

The *elderly and disabled assistant agent* (EDA) specializes in the mobility requirements and needs of any type of mobility impairment. The *EDA coalition creator* (EDAC) agent accepts requests aimed for EDAs and dynamically forms the coalition of different EDAs that will have to deliberate on the user's goal. The EDAC also acts as a provider agent because it advertises its service profile to the broker.

Figure 1 depicts the overall system architecture as a UML deployment diagram. The small people represent agent software modules and entities.

The agents serve the user on the basis of context, which relates to this information:

- sensor information (which the PWDA monitors), such as user heart rate, humidity, and lighting conditions;
- local services (which the AESA monitors), such as whether the user is indoors or outdoors, at home, or in a car, as well as public transport terminal local area network information;
- the time; and
- the weather report or 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 we present in detail elsewhere.[7]

Development of the system involved these main technologies:

- the Knoplerfish OSGi framework (for integrating the MAS to the other participating software modules on both the nomad device and the server subsystems; see www.knopflerfish.org),
- the JADE (Java Agent Development framework) agent platform (for developing the agents; see http://jade.tilab.com),
- the Jena Semantic Web framework (for semantic matchmaking by the broker agent; see http://jena.sourceforge.net), and
- the Protégé ontology editor (for developing the relevant domain ontology; see http://protege.stanford.edu).

For more information regarding the overall system architecture and development, see our previous work.[3]

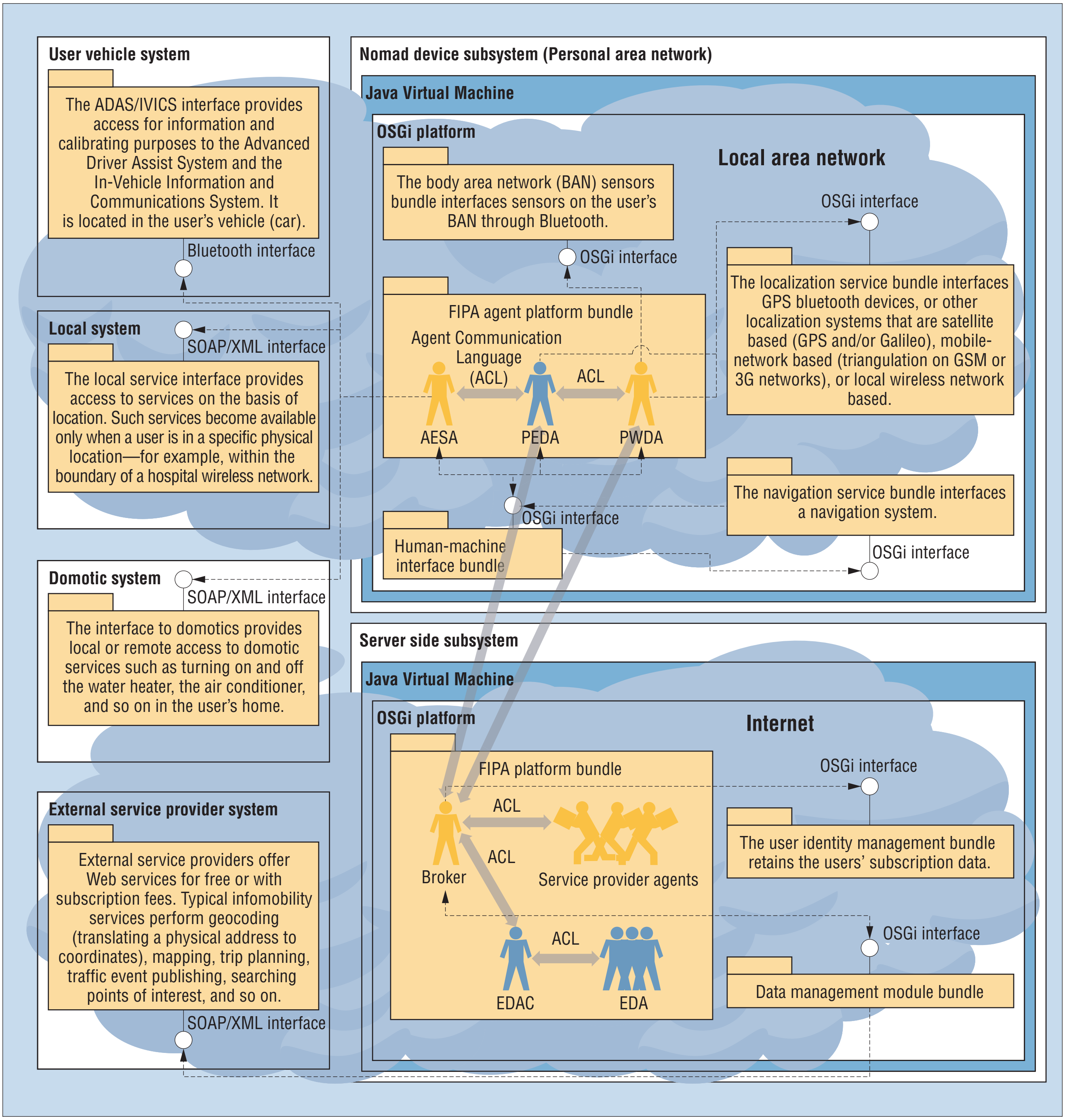

Figure 1. The system's architecture in the form of a UML deployment diagram. The people represent agent software modules and entities, including the ambient intelligent service agent (AESA), the personal wearable intelligent device agent (PEDA), the personal wearable communication device agent (PWDA), the elderly and disabled assistant agent (EDA), and the EDA coalition creator (EDAC).

The argumentation-based communication framework

We applied the communication framework that Antonis Kakas, Nicolas Maudet, and Pavlos Moraitis proposed,[8] which is based on Kakas and Moraitis's argumentation framework.[6] The communication framework adopted Peter McBurney and Simon Parsons's idea[9] that we can distinguish three languages in the representation of an agent's communication theory:

- $\mathcal{L}$, a language for describing the background information that the agent has about its world at any given moment and the basic rules for deciding its communication moves;

- $\mathcal{ML}$, a language for expressing preference policies pertaining to the decisions it makes during these moves; and
- $\mathcal{CL}$, a common communication language for all agents.

So, we define an argumentation theory 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 framework's background monotonic logic* ($\mathcal{L}$, $\vdash$), *defined as* $L \leftarrow L_1, \ldots, L_n$, *where* $L, L_1, \ldots, L_n$ *are positive or explicit negative ground literals. Rules in* $\mathcal{P}$ *are defined in the language* $\mathcal{ML}$, *which is the same as* $\mathcal{L}$ *except 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 must 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—that is, $T \vdash L$ under the background logic. The subset of rules in the argument T that belong to $\mathcal{T}$ is called the *object-level* argument. In general, we can separate out a part of the theory $\mathcal{T}_0 \subset \mathcal{T}$ and consider it a non-defeasible part from which any argument rule can draw information that it might need. We 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 (attack notion). *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) *if and only if there's a literal* L, $T_1 \subseteq T'$, $T_2 \subseteq T$, $P_1 \subseteq P'$ *and* $P_2 \subseteq P$ *such that*

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

Here, $S \vdash_{min} L$ means that $S \vdash L$ and that no proper subset of S implies L. Also, when L doesn't 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 counterargument to another such argument when it derives a contrary conclusion, L, and ($T' \cup P'$) makes the rules of its counterproof at least as strong as the rules for the proof by the argument under attack. The attack can occur on a contrary conclusion L = h-p(r, r′) that refers to the priority between rules.

Definition 3 (admissibility notion). *Let* ($\mathcal{T}$, $\mathcal{P}$) *be a theory,* $T \subseteq \mathcal{T}$ *and* $P \subseteq \mathcal{P}$. *Then* (T, P) *is admissible if and only if* ($T \cup P$) *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 if and only if* L *holds in a (respectively every) maximal (with respect to set inclusion) admissible subset of the theory.*

We have three levels in an agent's argumentation theory: object-level rules, default context priorities rules, and specific context priorities rules.

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

We define an agent's argumentation theory as a theory ($\mathcal{T}$, $\mathcal{P}$), in which $\mathcal{P}$ is separated into two parts:

Definition 4. *An agent's argumentative policy theory,* T, *is a theory* $T = ((\mathcal{T}, \mathcal{T}_0), \mathcal{P}_R, \mathcal{P}_C)$ *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 the head* h-p(r_1, r_2), *such that* $r_1, r_2 \in \mathcal{T}$ *(h-p standing for higher priority); and*
- *all rules in* $\mathcal{P}_C$ *are also priority rules with the head* h-p(R_1, R_2), *such that* $R_1, R_2 \in \mathcal{P}_R \cup \mathcal{P}_C$. $\mathcal{T}_0$ *contains auxiliary rules of the agent's background knowledge.*

So, we have three levels in an agent's theory. In the first, we have the rules $\mathcal{T}$ that refer directly to the agent's subject domain. We call these the agent's *object-level decision rules*. In the other two levels, we have rules that relate to the policy under which the agent uses its object-level decision rules associated with 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.

To explain how this argumentation framework works, we present an example where the theory $\mathcal{T}$ represents part of the object-level decision rules of a company employee (nonground rules represent their instances in a given Herbrand universe). Here and later, we use logic-programming notation, in which any term starting with a capital letter represents a variable. Abusing this notation, we'll denote the constant names of the priority rules R and C with capital letters.

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, a theory $\mathcal{P}_R$ represents the general default behavior of the company's code of contact in relation to its employees' roles. That is, a request from a superior is generally stronger than an employee's own need, and a request from another employee from a competing department is generally weaker than an employee's 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 the employee has an object and needs it urgently, then he or she 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)

We now present the communication framework that we used.[8] Agents interact using *dialogue moves* or *locutions*. The agents' shared communication language ($\mathcal{CL}$) contains a set of locutions of the form P(X, Y, C), where

- P is a performative type belonging to the set *Perf*,
- X and Y send and receive the locution, respectively, and
- C is the locution's content (that is, its body).

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

In the framework that Kakas, Maudet, and Moraitis proposed,[8] the agents' communication theory has three parts:

- The *basic component*, T_{basic}, defines the private dialogue steps.
- The *tactical component*, $T_{tactical}$, defines a private preference policy of tactics.
- The *attitude component*, $T_{attitude}$, captures general (application-independent) characteristics of the agent type's personal strategy.

We call $T_{basic} \cup T_{tactical}$ the *tactical theory* and $T_{basic} \cup T_{attitude}$ the *attitude theory*. Here, 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 the *enabling conditions* of the dialogue step from the performative p_i to p_j. In other words, these are the conditions under which agent X (whose theory this is) may utter p_j upon receiving p_i from agent Y.

The agent's background knowledge also contains 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, to express the general requirement that two different utterances are incompatible. 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 one. So, any admissible set of arguments can't contain rules that derive more than one utterance.

The tactical component defines a private preference policy that captures the agent's "professional" tactics for deciding among the alternatives enabled by the basic part of the theory. It consists of two sets $\mathcal{P}_R$, $\mathcal{P}_C$ of priority rules.

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

Knowledge regarding the needs of different mobility impairment types can be vast. It can also sometimes conflict if it tries to encompass all impairment types.

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

where SC_{jk} are specific conditions that are evaluated in the agent's background knowledge base and could depend on the agent Y, the incoming locution's content, and the types j and k of these alternative locutions. The first rule expresses the default preference of responding with p_k over responding with p_j, whereas the second rule states that under some specific conditions the preference is the reverse. 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, then, we can discriminate between the dialogue locutions by simply specifying that the agent prefers its default behavior unless certain conditions are specified. The latter 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.

When the special conditions hold, dilemmas (nondeterminism) 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 to resolve these conflicts and give priority to rules with the special conditions. So, we can have rules of the form

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

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

Argumentation-based agent interaction

Knowledge regarding the needs of different mobility impairment types can be vast. It can also sometimes conflict if it tries to encompass all impairment types. So, we decided to define it separately for each impairment type. Then, for a person with more than one type of impairment, a coalition of agents—each with knowledge of a different impairment type—will deliberate on the user's needs. These EDAs interact through an argumentation-based communication protocol. Thus, whenever the user wants to plan a trip, the PEDA adds the context information and the user profile data relevant to the request and sends it to a broker agent that forwards it to an EDAC agent. The EDAC agent is responsible for finding a group of EDAs, each an expert in one of the user's impairments. Then, the server-side EDAs deliberate to define the characteristics of the transportation mean that satisfies all the user's impairments. The EDAs send this dialogue's result back to the EDAC agent, which prepares a request for the broker. The latter finds the appropriate route-calculating service providers that can service the precisely defined request (for example, containing the desired transportation means).

The FIPA interaction protocol instance (service protocol) in figure 2 shows this sequence. A PEDA asks for a trip-planning infomobility service and includes information about the user's context. The broker agent matches the request to its service profile repository and finds the relevant advertisement submitted earlier by the EDAC agent, to which it forwards the request. The EDAC agent extracts the user impairment types from

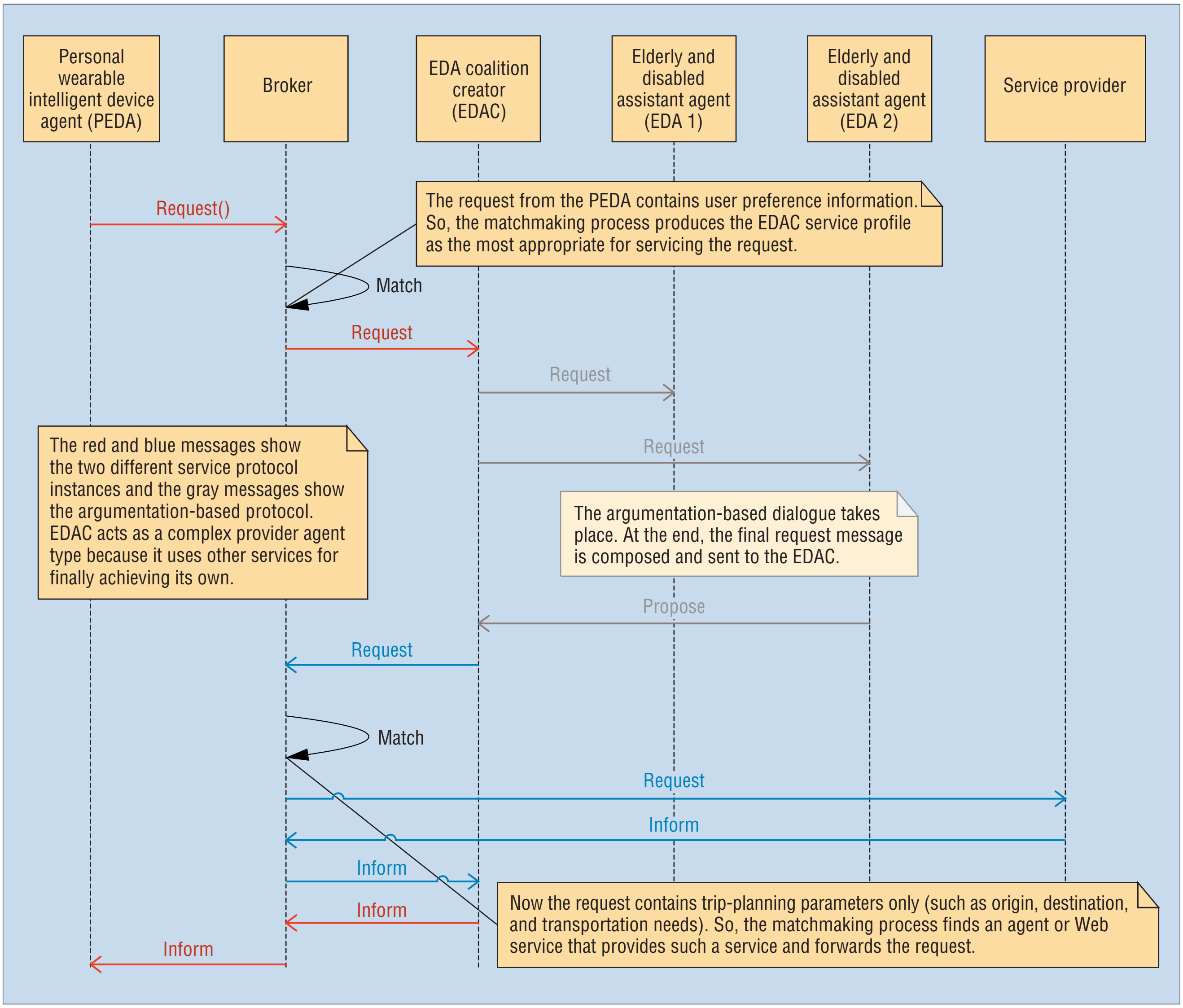

Figure 2. The argumentation dialogue in the context of service provisioning.

the user profile part included in the request and determines which EDA should be involved in the coalition that will determine the user's trip-planning needs. Then, it sends a request to these EDAs, including the list of EDA participants and the original PEDA request. Subsequently, the argumentation dialogue takes place. It ends when all EDAs agree on a proposal. The EDA that originally issued the agreed-upon proposal forwards it to the EDAC, which integrates it in the original PEDA request and removes all user preference, profile, and context data. Then, it sends this new request to the broker agent, which 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, it forwards them to the EDAC. Then, the EDAC replies to the broker's first request, so the original service protocol instance is also completed. Finally, the broker forwards the results to the PEDA. 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.

This argumentation-based communication protocol is based on the theoretical work we presented in the argumentation-based communication framework section. The strategies that relate to the agents' expertise and define their behavior in the distributed decision-making process are represented as argumentation theories. These theories have two components: the first contains object-level ($\mathcal{T}$) rules, and the second contains default ($\mathcal{P}_R$) rules and specific context ($\mathcal{P}_C$) rules. The first component rules represent the agents' possible dialogue moves. The second component rules 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. The possible replies to a received message are determined by its performative. The choice of the reply message type that will be sent (for instance, argue, accept, or challenge), among several

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

Figure 3. The argumentation theory of the wheelchair agent.

possible candidates, and the appropriate content are 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. It represents the best decision that an agent can make in a specific step of the dialogue, according to its theory (that is, its expertise) and the information it has exchanged with the other agents up to that step.

Our system currently allows only bilateral dialogues, so if a user has more than two impairments, several successive bilateral dialogues will occur. When two experts reach an agreement, the one whose proposal the other accepted will represent both experts in a new bilateral dialogue with a third expert.

We implemented the argumentation-based agent interaction using Gorgias (an open source software implementation of Kakas and Moraitis's argumentation framework;[6] see www2.cs.ucy.ac.cy/~nkd/gorgias) and JADE.

Real-world scenarios

Here, we present two real-world scenarios depicting two use cases corresponding to the application requirements. For space and comprehensibility, we present simplified versions of these scenarios and therefore only a part of the theories that represent the involved agents' knowledge. So, the dialogue's only goal is to agree on what means of transportation the user should select for his or her trip. In more complex scenarios, the agents must agree on more issues, such as

- the maximum number of meters that the user should walk,
- his or her preference between the fastest trip or one with fewer connections, and
- his or her needs regarding human assistance and roadside services (such as toilets).

Scenario 1

Consider a person who uses a wheelchair and has heart problems. When he's moving in the city, he always prefers to use public transportation (such as a bus or the metro) for economical reasons. In particular, he prefers the bus because the access is easier than the metro and he can enjoy the itinerary. However, because of his heart problems, he must sometimes take the metro, especially when the weather is bad.

Two different expert agents—a wheelchair expert (WA) and a heart problems expert (HPA)—take these two transportation preferences into account. Both experts consider the user's preference for the bus (their background knowledge includes this information). Only in particular situations would they propose another transportation solution—in this case, the metro. The WA proposes by default the bus as the most convenient (preferable) means of transportation. However, it can agree to a different transportation mean when particular situations arise related to heart problems or the weather (or both). The HPA needs to get the user's heart rate before advising, and it consults the weather prediction. It prefers by default the metro because the user is less exposed to the elements, thus decreasing his risk of catching a cold, which would be dangerous because of his heart problems. However, the HPA can accept another means of transportation (such as the bus when the weather is good or when the distance to cover is short, provided that it's not snowing). These agents' behaviors can be represented through the following strategies.

The wheelchair agent. When the user asks to plan a trip, the WA proposes by default the bus (rule r_1 in figure 3). When another agent proposes a different transportation means, the WA asks for an explanation (rules r_2 and R_1). However, if the distance is long, it accepts the other proposal (rules r_6 and R_3). When another agent asks the WA for an explanation about its proposal to use the bus, it gives the reason, which is that the user prefers the bus. In normal conditions, all agents accept this (rule r_2).

When another agent proposes the bus, the WA agrees (rules r_6 and R_2). When another agent proposes a different means of transportation and the WA agrees with its reasons, it agrees (rules r_4, R_5, R_6, C_2, and C_3). Otherwise, it sends a counterargument, if it has an admissible one. In this specific scenario, these arguments can be either "the bus is the most preferable means of transportation" or "the distance is short" (according to the situation), the latter being valid even if the weather is cold (rules r_7, R_4, and C_1). However, when the proposal concerns the taxi, the WA always agrees (rules R_6 and C_3).

When the WA receives another agent's agreement on its proposal to use the bus, it 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, the situation needing the taxi is considered mandatory, so the WA opts for the taxi (rules r_4, R_5, R_6, C_3, and C_4).

We represent these strategies by using the adopted argumentation framework through the theory in figure 3. To simplify the notation, we use r_1 instead of $r_{propose,\ request}$ for the object-level rules as would normally be the case following the notation we presented in the argumentation-based communication framework section. For the same reason, we use R_1 for the priorities rules instead of $R_{challenge\backslash agree}$. The same holds for all the other "r," "R," and "C" rules.

In this scenario, the WA agent has no knowledge about the weather or the user situation. So, before any interaction with the other agents that have these pieces of information, the WA evaluates the "abducibles" predicates[6] {nice_ weather(Day), ¬snowing(Day), normal_ heart _rate(Z)} to true. The equivalent holds for all the agents.

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

Figure 4. The argumentation theory of the heart problems agent.

The heart problems agent. When the user asks to plan a trip, the HPA proposes the metro if the user's heart rate is abnormal so that the user is less exposed to cold or warm weather (rule r_1). When another agent proposes a different means of transportation, the HPA asks for explanation if it can't accept this mean in the current situation (rules r_2, R_1, and C_6). Otherwise, it accepts immediately (rules r_6, R_6, and C_5). When another agent asks it for an explanation about its own proposal to use the metro, it gives the reason (rule r_3).

When another agent proposes a transportation means other than the metro and its reasons convince the HPA, the HPA agrees (rules r_4, R_4, R_5, C_1, and C_3). When another agent proposes the metro, the HPA agrees (rules r_6 and R_2). Otherwise, it sends a counterargument, if it has an admissible one (rules r_7, R_3, and C_2). In this scenario, these arguments can be "the weather is cold," "the user's heart rate is abnormal," or "it is snowing," according to the situation. However, when the proposal concerns the taxi, it always agrees (rules R_5 and C_3).

When the HPA receives another agent's agreement to its proposal to use the metro, it 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, the situation needing the taxi is considered mandatory, so the HPA opts for the taxi because the taxi isn't dangerous for the heart problems (rules R_3, R_5, and C_3). The same holds when bus and taxi are both under consideration (rules R_4, R_5, and C_4).

This agent's default behavior is to counterargue and to accept another proposal only if specific situations hold (see rule R_3). The theory in figure 4 represents these strategies.

An example dialogue. Let's consider a situation where the person using a wheelchair and having an abnormal heart rate (which is part of the HPA's background knowledge) wants to move from his house to his mother's house. The distance between the two places is short (which is part of the WA's background knowledge). The weather is cold, and it's snowing (which is part of the HPA's background knowledge). This situation might generate this dialogue between the agents (presented graphically in figure 5):

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, and R_1)
WA: Why the metro? (rules r_2, r_6, and 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, and C_8)
WA: But the distance is short (rules r_4, r_7, R_4, R_5, and C_1)
HPA: Yes, but it is snowing (rules r_4, r_7, R_3, R_4, and C_2)
WA: OK. I agree (rules r_4, r_7, R_4, R_5, and C_2)

Scenario 2

Now consider a situation where a user has heart and physical problems. In this situation, a new expert, the *physical situation agent* (PSA), which cares about the user's physical situation, must be involved. This agent can accept any means of transportation provided that the user doesn't feel tired. If this is the case, the PSA agent can't accept any other means of transportation than the taxi.

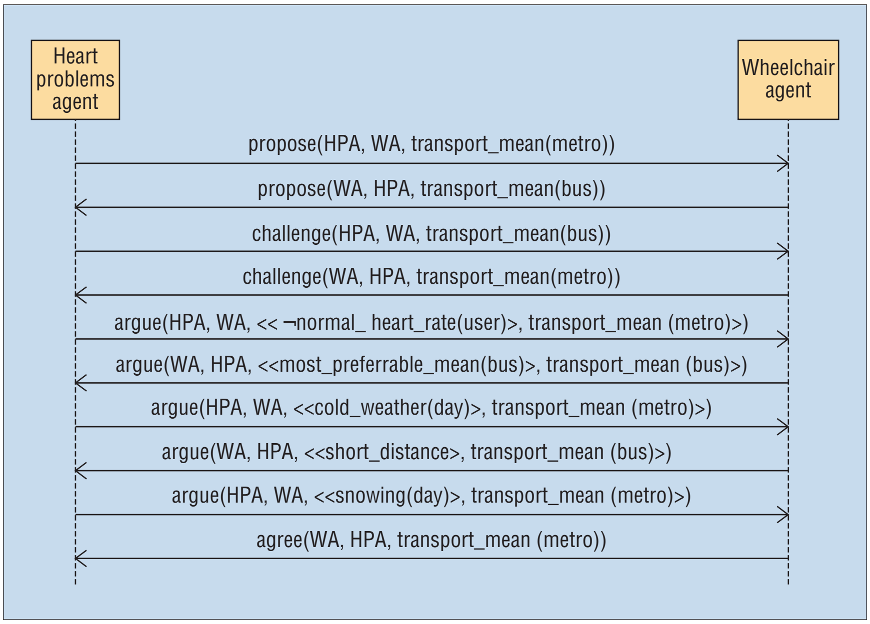

Figure 5. The AUML (agent-based unified modeling language) sequence diagram for the first scenario.

r_1: propose(X, Y, transport_mean(taxi)) ← request(Z, X, move(Place, Place′)), feel_tired(Z)
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(Z)
R_4: h-p(r_5, r_2) ← transport_mean(metro), ¬feel_tired(Z)
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)}

Figure 6. The argumentation theory of the physical situation agent.

When the user asks to plan a trip, the PSA makes a proposal. Particularly, it proposes the taxi when the user feels tired (rule r_1). In this situation, the PSA can accept only this solution (rules r_2, r_5, and R_1). When another agent proposes the taxi, it agrees (rules r_2, r_5, R_2, and C_1). Otherwise, if the user isn't feeling tired, the PSA can accept the metro or the bus (rules r_2, r_5, R_3, R_4, C_2, and C_3).

When another agent asks the PSA for an explanation about its proposal to use the taxi, it gives the reason (rule r_3). When it receives another agent's agreement on its proposal to use the taxi, it sends this proposal to the coalition creator. This declares the termination of the dialogue. The theory in figure 6 represent these strategies.

Suppose the user feels tired, has an abnormal heart rate, and wants to move from his house to his girlfriend's house. This situation might generate this dialogue between the HPA and PSA:

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, and C_3)

To our knowledge, this is the first work to use argumentation in such a real-world ambient-intelligence application. This prototype shows that argumentation is well suited for implementing complex preference-reasoning mechanisms at the individual-agent level and for implementing complex interaction protocols where high-level agent dialogues can take place. So far, we've tested only trip-planning scenarios.

Our future research will concern modeling dialogues where more than two agents can participate simultaneously. We also plan to enhance the agents' theories (and therefore the protocol) for deliberating about other types of services than travel services.

Acknowledgments

We thank the reviewers for their valuable, constructive comments and Singular Logic S.A. (Greece) and the ASK-IT consortium for supporting this work. The ASK-IT (IST-2003-511298) IP provided the requirements and partial funding for this work (see www.ask-it.org.).

References

1. S. Poslad et al., "CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism," *Proc. 2nd Int'l Conf. 3G Mobile Communication Technologies*, IET Conf. Proc., 2001, pp. 28–32.
2. P. Moraitis, E. Petraki, and N. Spanoudakis, "An Agent-Based System for Infomobility Services," *Proc. 3rd European Workshop Multi-Agent Systems* (EUMAS 2005), Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, 2005, pp. 224–235.
3. N. Spanoudakis and P. Moraitis, "Agent Based Architecture in an Ambient Intelligence Context," *Proc. 4th European Workshop Multi-Agent Systems* (EUMAS 06), 2006, pp. 163–174.
4. *Argumentation in Multi-Agent Systems*, LNCS 3366, Springer, 2005.
5. *Argumentation in Multi-Agent Systems*, LNCS 4049, Springer, 2006.
6. A. Kakas and P. Moraitis, "Argumentation Based Decision Making for Autonomous Agents," *Proc. 2nd Int'l Joint Conf. Au-*

tonomous Agents and Multi-Agent Systems (AAMAS 03), ACM Press, 2003, pp. 883–890.

7. N. Spanoudakis and P. Moraitis, "Engineering a Brokering Framework for Providing Semantic Services to Agents on Lightweight Devices," *Proc. ECAI 06 Workshop Context and Ontologies: Theory, Practice and Applications* (C&O 06), Università di Trento, 2006, pp. 56–58.
8. A. Kakas, N. Maudet, and P. Moraitis, "Modular Representation of Agent Interaction Rules through Argumentation," *J. Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 2, 2005, pp. 189–206.
9. P. McBurney and S. Parsons, "Argumentation-Based Communication between Agents," *Communication in Multi-Agent Systems,* LNAI 2650, M.P. Huget, ed., Springer, 2003.

The Authors

Pavlos Moraitis is a professor in the Department of Mathematics and Computer Science at Paris Descartes University. His main research interests include argumentation, automated negotiation, agent communication languages and dialogues, multiagent planning, automated decision making, agent architectures, and agent-oriented software engineering. He received his PhD in computer science and artificial intelligence from Paris Dauphine University. Contact him at the Dept. of Mathematics & Computer Science, Paris Descartes Univ., 45, rue des Saints-Pères, 75270 Paris Cedex 06, France; pavlos@math-info.univ-paris5.fr.

Nikolaos Spanoudakis holds a teaching assistant position in the Applied Mathematics and Computers Laboratory at the Technical University of Crete's Department of Sciences. He's also a PhD candidate in the area of intelligent agents and multiagent systems in Paris Descartes University's Department of Mathematics and Computer Science. His main research interests are agent-oriented software engineering, agent architectures, and applications of multiagent systems in ambient intelligence. He's a member of the IEEE, the International Association of Software Architects, the Greek Computer Society, and the Technical Chamber of Greece. Contact him at the Dept. of Sciences, Technical Univ. of Crete, Univ. Campus, Kounoupidiana, 73100, Chania, Greece; nikos@science.tuc.gr.