An architecture for negotiation with mobile agents

Jaime Delgado[1] , Isabel Gallego[2] , Roberto García[1] , Rosa Gil[1]

[1] Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{jaime.delgado, roberto.garcia, rosa.gil}@tecn.upf.es
[2] Universitat Politècnica de Catalunya, Departament d'Arquitectura de Computadors,
Campus Nord, Mòdul D6, Jordi Girona 1-3, E-08034 Barcelona, Spain
isabel@ac.upc.es

Abstract. New applications make networks increase their traffic daily, so a reliable and robust design must be provided to conform an architecture that could satisfy new needs. Terms as knowledge management, agents, ontologies, are key issues to achieve that architecture. Agent technology allows us to carry out interoperability and autonomy among the modules that build the architecture. Different kinds of devices are appearing, such as wireless devices. Then, a new group of applications are coming into the market, and a relevant characteristic is common in many cases: mobility. The general architecture for mobile agents applications we are describing in the paper generalises a set of applications and tools we are developing. We start the work in the context of Digital Rights Management (DRM), and a specific negotiation scenario and its implementation with mobile agents is discussed, before giving technical details of the architecture. This generic architecture includes elements coming from different disciplines, that are assembled to form a structure that fits diverse environments.

1 Digital Rights Management

A very promising sector for the application of mobile agents is e-commerce, in particular the e-commerce of multimedia content.

When trading with multimedia content, one of the key issues is what does it happen with the Intellectual Property Rights (IPR) that are normally associated to that content, being a picture, a song, a video clip or any other audiovisual content.

When managing those IPR, what is normally called Digital Rights Management (DRM), we have to face with many issues, most of them not having currently a really optimal solution.

We first developed a system for e-commerce of video content (MARS project [1]), in which we initially focused on data localisation aspects following different metadata techniques for its implementation. Then, we added copyright information to that metadata, and we included watermarks in the content itself. The results of this project were our main starting point for the work presented in this paper.

Apart from metadata, in relation to representation of the information, there are other aspects to consider when dealing with DRM, such as the issues associated to the

Rights Expression Languages (REL) and Rights Data Dictionaries (RDD). A Rights Expression Language is seen as a machine-readable language than can declare rights and permissions using the terms as defined in a Rights Data Dictionary that provides a set of clear, consistent, structured and integrated definitions of terms for use in the REL. For this, the current standardisation efforts in MPEG-21 [2], that try to integrate previously existing initiatives, are a clear example.

A different approach to that problem consists on specifying an IPR ontology able to add semantic information for the management of digital rights, allowing in this way the interoperability of different applications. We have developed "IPROnto" [3], an IPR ontology that formalises the IPR domain with a Semantic Web approach, describing IPR contracts, actors, intellectual property creations, rights, etc., together with a complete metadata framework and a rights data dictionary, using RDFSchema, DAML+OIL, …

1.1 IPR Business Models

In order to work on DRM, we need to select a model in which to base IPR representation and negotiation. The IMPRIMATUR Business Model [4], the one we have selected, identifies a series of entities that may take different roles, such as Creator, Provider, Rights Holder, Distributor, IPR Data Base, or Watermarking & Fingerprint marking.

A simplified and specific model, the one we are implementing, consists on the use of a Broker (with the role of Distributor) in charge of being an intermediary between providers of multimedia material (content providers) and customers interested in buying that material and the corresponding rights for use and/or commercial exploitation. From a functional point of view, these copyrighted multimedia material providers may also assume the roles of Creator and Rights Holder in the same entity.

Furthermore, the broker stores and keeps up to date (with the help of content providers) the information about the multimedia material for sale in the system (from all content providers associated to the broker), and about the terms and conditions in which commercial electronic transactions are done, with the help of the IPR Database. Fig. 1 illustrates this Broker Based IPR General Model.

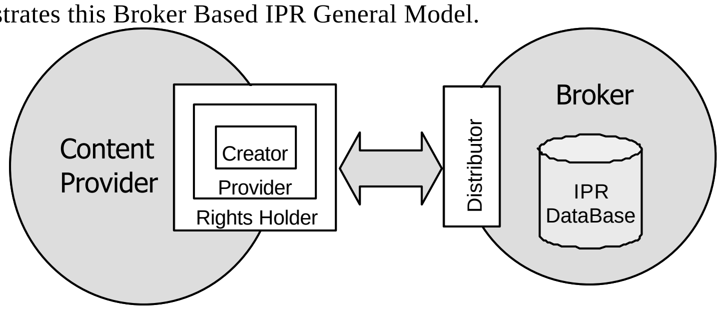

Fig. 1. Broker Based IPR General Model

1.2 Negotiation of IPR conditions

Based on the IPR attributes set, when a buyer requests, to the broker, a purchase of audiovisual material subject to copyright, the broker extracts IPR information from its database and presents an initial offer to the buyer. This information allows the buyer to take a decision on how to buy IPR, i.e., to know what are the copyright rules associated to the asset, to decide if to re-sell it, etc. To facilitate this process, a negotiation mechanism has developed, and it is described later.

The negotiation protocol, that it is part of the “Service Request” phase in an e-commerce model [5], has three sub-phases:

– Initial offer,
– Co-operative contract production, and
– Payment.

In the Contract production sub-phase, the most complex and important one, there are several alternatives over which to work. First, the selling entity initiates the protocol with an initial proposal of digital rights conditions, normally taken from a pre-defined subset. After that, the buying entity has three alternatives:

1. Accepting the offer,
2. Making a counter-offer and
3. Rejecting the offer.

After the initial proposal, the negotiating entities elaborate the contract, using the negotiation protocol, from the sequence of offers and counter-offers until a final agreement is reached, forming then the final electronic contract.

2 An implementation with mobile agents

We are implementing our agents and ontology framework, described in this paper, in the context of the NewMARS project, an extension of the MARS project [1], mentioned before, and integrated as a subproject of the AREA2000 project[1]. Apart from the results of the MARS project, we are also using results from other previous work [6] in our research group.

As an example, we describe how we implement a specific DRM negotiation scenario, taken from MPEG-21 work [7]. In the scenario, the user is a web designer that has decided to use a specific image in her current web work. She wants to locate a specific version of this image and acquire the necessary digital rights to use it.

The phases of this scenario, according to the NewMARS implementation, are: user interaction, search, negotiation, outcome presentation and control. A sequence diagram of them, except for the background control phase, is shown in Fig. 2.

To fulfil this scenario, the web designer of our example uses the NewMARS facilities, so she interacts locally with her NewMARS user agent residing in her mobile device. She uses agent’s GUI to determine all the criteria required to select the image she is interested in. Then, the agent enters the search phase and it migrates to another agent-platform container (section 3.2 explains this process and why it is so

[1] Project supported by the Spanish government (TIC2000-0317-P4-05).

important). This new container is one of the server kind, where it has better Internet connectivity and processing power. Thanks to these greater resources, it can carry on less constrained searches and a more accurate negotiation process of the required image.

The search is performed through the NewMARS meta-search engine [8] that looks for the required image in some Internet image directories. When the content is found, the meta-search agent returns a reference to the agent managing the image rights and the image identifier.

When the user agent owns enough information, it can start the automatic negotiation process through a protocol defined by rules (see section 3). The retrieved licensing agent is contacted and a call for proposals is issued. An initial offer is received, if the licensing agent really has the requested image. From this point, some counter-offers may be interchanged till the negotiation ends due to a reject or an agreement.

The negotiation results are then communicated to the user. To facilitate user interaction, the user agent returns to the agent-platform container at the users mobile device. In parallel, a post-agreement background phase is initiated in the NewMARS system. This control phase is conducted to guarantee a fair use of the negotiated product.

A more detailed description of these phases is given in the following sub-sections.

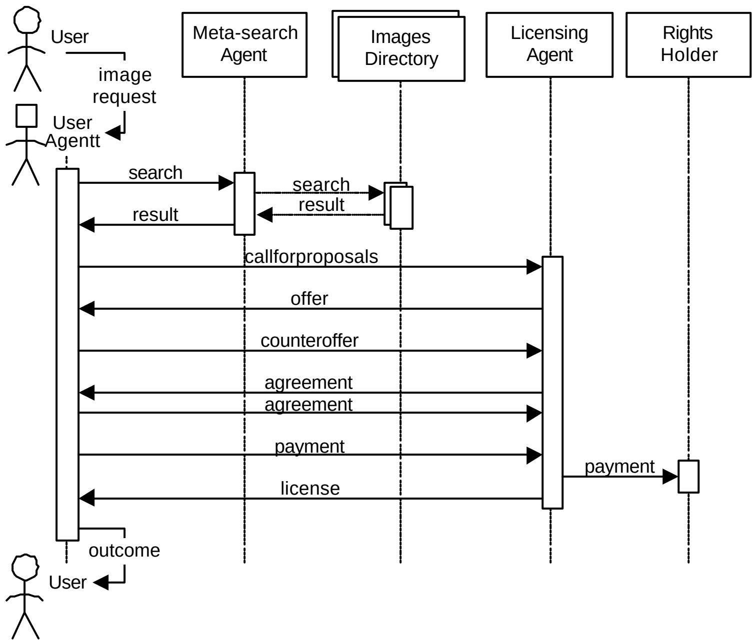

Fig. 2. Sequence diagram of the negotiation NewMARS scenario

2.1 User interaction

The user initiates her user agent for images search and negotiation. This can be done in her desktop or, in our scenario, in her mobile device running Personal Java. The GUI shows up and the user interacts with the forms shown to specify the image she is interested in and in what conditions.

The GUI allows defining image characteristics, like image format, title, subject, size, dimensions, etc. The user can also determine the negotiation conditions, like price, allowed uses, period of time, etc. During the interaction, the user agent checks the defined image properties and values against the used ontologies, one for images and another one for digital rights. Thus, only valid constraints can be defined and the agent can assist the user during their definition. The constraints are modelled using RDF Model and Syntax [9] and the ontologies against which they are checked are modelled using RDF Schema [10]. RDF defines the syntax and semantics of the exchanged messages. An example is given in Table 1.

Finally, the user submits the checked search and negotiation conditions. The user agent internally stores them and enters the search phase, detailed in the next section.

2.2 Search

When the user agent enters into the search phase, it moves from the lightweight container in the user wireless device to a server container in a wired and more powerful machine. From this new location the user agent contacts a directory of services to locate the meta-search engine. This directory is initially implemented using the FIPA [11] Directory Facilitator (DF) provided with the FIPA-compliant agent platform that is used. The DF returns a pointer to the agent implementing the meta-search engine, which previously registered itself in the DF.

Once the user agent has located the meta-search agent, it can use its predefined interaction protocol. A FIPA ACL message containing the image characteristics and negotiation constraints is sent. The meta-search agent processes it and, after performing the necessary searches, returns a set of locations of licensing agents. They have registered themselves as negotiators of the requested image. One of them is selected and the negotiation process starts.

2.3 Negotiation

Once the user agent has selected a reference to a provider of the image it is looking for, the negotiation to obtain it begins. The negotiation protocol is obtained from the agent platform, where it has been previously registered.

First, the customer agent issues a call for proposals referred to the desired image. Then, the licensing agent responds with an initial offer if it has the requested content, a refusal otherwise. An example of offer is shown in Table 1. Given that we are considering a totally automatic scenario, the user agent analyses this offer and decides what to do afterwards. If it does not accept the offer conditions, it can formulate a counter-offer.

Table 1. Example of offer serialised as RDF/XML

```
<rdf:RDF   xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:ipr  = "http://dmag.upf.es/ontologies/ipronto#"
           xmlns:xsd  = "http://www.w3.org/2000/10/XMLSchema#">
<ipr:Offer rdf:about="http://dmag.upf.es/mars/offer20020211183424">
  <ipr:patient>
  <ipr:PurchaseLicense>
    <ipr:licenser rdf:resource="http://www.howlinwolf.com"/>
    <ipr:licensee rdf:resource="http://chiTouristGuide.org"/>
    <ipr:permision> <ipr:Access>
        <ipr:place rdf:resource="http://chiTouristGuide.com/issues/march02"/>
        <ipr:patient rdf:resource="http://www.howlinwolf.com/imgs/0973.jpg"/>
        <ipr:user rdf:resource="http://chiTouristGuide.com/members"/>
        <ipr:timeFrom> <xsd:date rdf:value="2002-03-01"/> </ipr:timeFrom>
        <ipr:timeTo> <xsd:date rdf:value="2003-03-01"/> </ipr:timeTo>
    </ipr:Access> </ipr:permission>
    <ipr:obligation> <ipr:Compensation>
        <ipr:payer rdf:resource="http://chiTouristGuide.com "/>
        <ipr:payee rdf:resource="http://www.howlinwolf.com"/>
        <ipr:input><ipr:DollarQuantity rdf:value="100"/></ipr:input>
    </ipr:Compensation> </ipr:obligation>
    <ipr:time>  <xsd:date rdf:value="2002-02-11"/> </ipr:time>
  </ipr:PurchaseLicense>
  </ipr:patient>
</ipr:Offer>
</rdf:RDF>
```

The same applies for the licensing agent when it receives the counter-offer. This interchange of counter-offers continues till any of the parties abandons the negotiation or an agreement arises.

Finally, when any of the parties agrees with the last offer, the other party can also agree and an agreement is reached. An electronic contract is produced, i.e. a RDF/XML document quite similar to that shown in Table 1. It contains the agreed conditions and two extra elements pointing to both license consenters. Both parties digitally sign it with XML Signature [12] and the result is a license that authorises the costumer to use the negotiated content under the stated conditions [13].

2.4 Outcome presentation and control

When the negotiation has finalised, the results are communicated to the user. To facilitate user interaction, the user agent returns to the agent-platform container at the users wireless device.

If the user agent has succeeded, a RDF document representing the achieved agreement is presented to the user through the agent GUI. It is formatted in a user-friendly manner and it contains a URL pointing to a location from where the licensed image can be retrieved.

All the involved parts have digitally signed the RDF agreement; really their representative agents have performed this action. Therefore, it can be used as a licensing contract that proves that the requested uses have been authorised.

If not agreement has been achieved the user agent presents the last outcomes of the negotiation process, the offer, counteroffer or refusal where the negotiation broke.

Finally, the control phase occurs in parallel and continues in the background. Its purpose is to monitor that the customer fulfils the conditions established in the agreement license. In [6] we outlined different approaches to this issue using mobile agents.

3 The architecture

Fig. 3 outlines the architecture we are developing. This is more general than the DRM negotiation implementation we have described in the previous section, since some parts still need to be developed further. In the following, we detail some technical aspects of the agents' architecture we have designed.

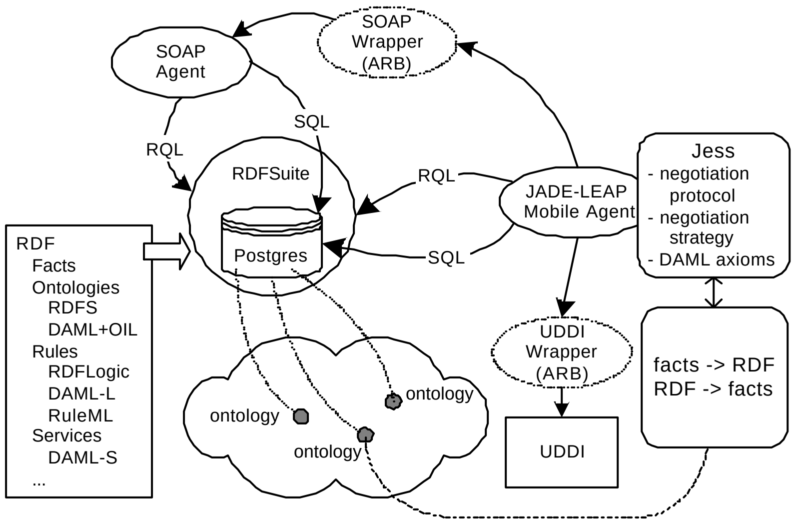

Fig. 3. Model Architecture

The managed data, the facts, are stored in a database using ICS-FORTH RDFSuite [14] that provides persistence and a RDF query language, RQL.

The RDF facts are structured using different ontologies, allowing to model processes and rules, like DAML-S [15], RDFLogic [16] or RuleML [17]. DAML+OIL [18] capabilities are planned using Prolog and DAML+OIL axioms [19].

Agents are modelled using JADE [20], based on FIPA Standard [11]. They are coordinated by an inference engine, in our case Jess [21], that also cares about the negotiation protocol.

CLIPS files (the Jess expert system format) store rules that govern coordination and negotiation protocol besides rules that allow understanding DAML ontologies.

The coordination role consists on controlling invocations of JADE agents and their association to negotiation protocols by loading the corresponding CLIPS file that contains them. This file manages negotiation protocols, so it takes cares of message interchanging and translating the RDF content to Jess facts. Translation is performed by a DAMLJessKB module [22]. This can be done because Jess has DAML axioms as Jess rules, which represent an ontology inside this data base. Jess knowledge base interprets these facts firing rules and generating a result that will be converted back to RDF or DAML.

We will concentrate on the final part, where the connection between agents belonging to a JADE platform and final users, who are represented by mobile agents inside a wireless device.

3.1 Mobile agents for wireless devices

In February 2002, a new agent platform implementation appeared, JADE-LEAP[20], designed to integrate wireless devices.

On the other hand, Sun [23] has developed J2ME, intended to wireless devices too (see Fig. 4), where we can find MIDP profiles [24], which are a subset of PersonalJava [25], allowing to build more sophisticated applications because it has been thought for PDAs.

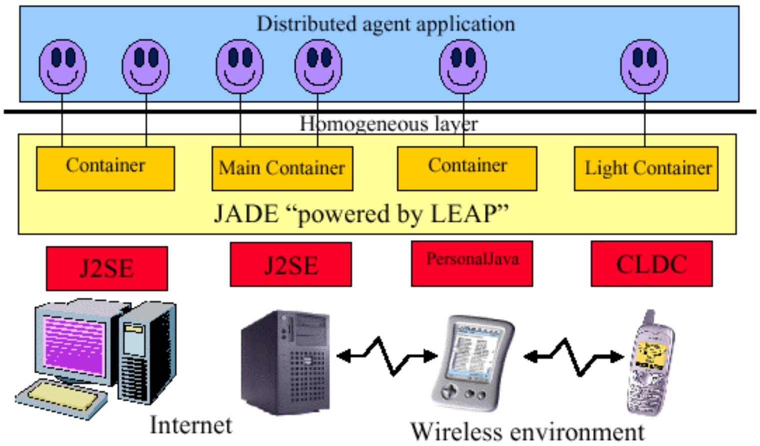

Fig. 4. JADE-LEAP agents architecture, from JADE-LEAP user's guide.

Also we find some restrictions, as GUI on PersonalJava. It must be developed with the standard Java AWT (Abstract Windowing Toolkit) and not Java Swing interface is possible. In every wireless device there is an AWT GUI (Graphical User Interface) associated to our agent that shows an agent list and its behaviour policies.

Agents and policies are mobile and they are moved when the user decides it, so the agents send a call to Jess in the main-container. In the next section we will see how Jess engine manages mobility.

3.2 Mobility policy

Mobility is managed from the Jess engine in the main container, a J2SE [26] one. It contains a set of rules, that depends on the specific application, that move agents when it is convenient and balance the load of the different containers.

As we can see in Fig. 5, a mobile agent has capability for moving from a wireless device, as iPAQ or PersonalJava, to another resource in the same platform (intra-platform mobility) to carry out a mission. It would be valuable when, for instance, a negotiation must be done, because it means a lot of messages going up and down. The process takes place in a J2SE container, in a local server, so Jess rules govern the negotiation protocol and its policy.

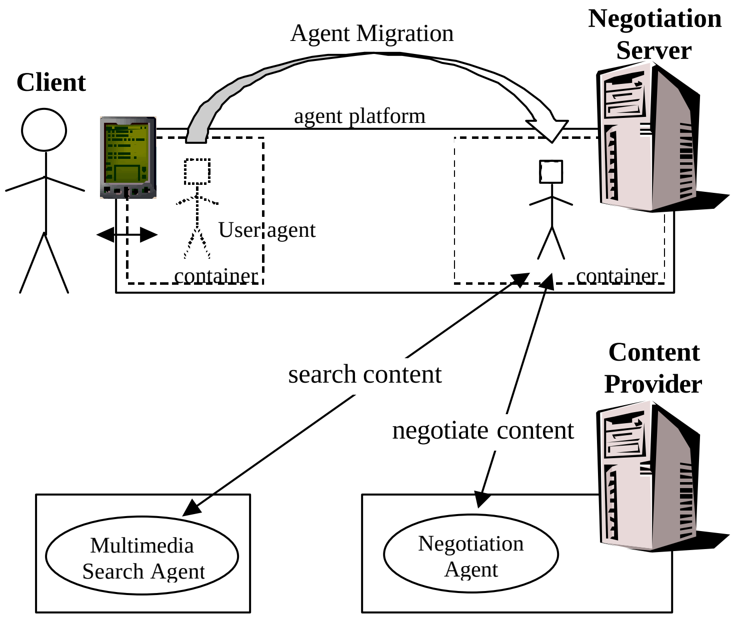

Fig. 5. Negotiation mobile-agent scenario

The Jess inference engine governs agents' behaviour, what includes their mobility patterns. User agents reside initially in the client hardware device, but these devices can have limited connectivity and processing power. When an agent wants to search and negotiate some content in the Internet, it is more efficient to do that from a wired and powerful device, what we have called a Negotiation Server.

Under these circumstances, the mobility behaviour rules are put into action. Fig. 6 shows an example mobility rule that is fired because its conditions are met Its complexity depends, again, on the specific application. When an agent is operating from a mobile location and willing to start a connection intensive interaction, if there is an available server location, it is moved from the mobile location to the server.

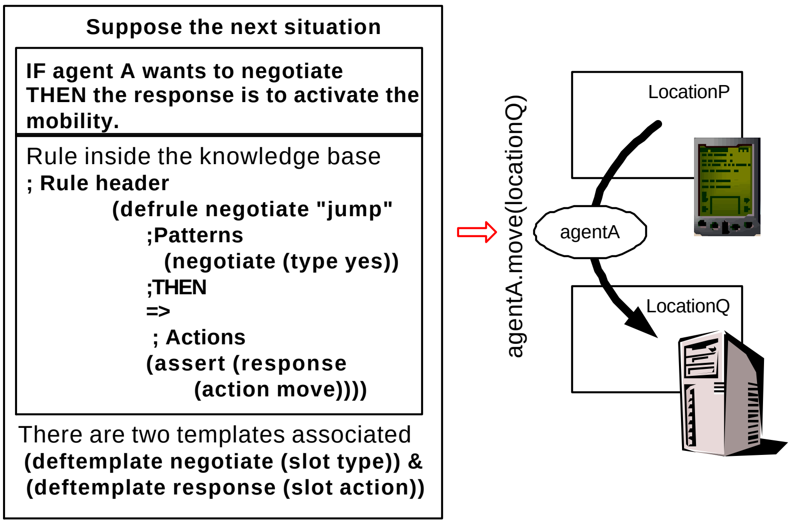

Fig. 6. Example of rule governing mobility

4 Future work and conclusions

We have started from the issue of Digital Rights Management, looking to solve the problem of negotiation of rights, that we approach with the idea of using mobile agents. A solution along these lines has been implemented in the mentioned NewMARS project. From this, we have developed a general architecture for mobility using agent technology, that has been described.

Our future work focuses on integrating all the presented architecture with the World Wide Web, more specifically with the Web Services environment. Web Services are self-contained programs and devices accessible from the web, for instance a web agent [27, 28]. They can be classified as information-gathering services, e.g. the map service at yahoo.com, or as world-altering services, e.g. the book buying service at amazon.com.

Web Services are based on three pillars. First, a directory of services where they are registered, UDDI [29]. Second, a service description language that allows automatic matching and invocation of services. Here the main alternatives are WSDL [30] and DAML-S [15]. And finally, a message transport service for Web Services interaction, SOAP [31].

All these Web Services tools will be incorporated to our current FIPA compliant agent architecture. UDDI will be a complement of the used FIPA Directory Facilitator. It will provide DAML-S services descriptions that will enable automatic location and invocation of services in the heterogeneous web domain. Moreover, interactions between agents and services using SOAP messages will overcome agent platform restrictions. The building blocks needed to integrate Web Services in our architecture are shown in Fig. 3 with thin-dotted lines. They are the SOAP and UDDI wrappers that allow interaction with UDDI registries and SOAP agents.

For instance, a DAML-S description of a licensing web service can be stored in the UDDI directory. Afterwards, it can be used to enrich query matching by the meta-search engine. Once located, the licensing service description is forwarded to the user agent, that uses the detailed description of the service to retrieve the negotiation protocol followed by the service. Thus, we are not constrained to previously established interaction protocols and the full dynamism of the agents and Web Services approaches can be exploited.

As a conclusion, new technologies are growing, so it is necessary to provide a way to interconnect them; this is the first step we want to achieve with the architecture. Mobility in this case allows us to negotiate in good conditions, which it is a basic premise because of negotiation complexity when new challenges as semantics and intelligence are involved.

References

1. Garcia, R.; Delgado, J.: Brokerage of Intellectual Property Rights in the Semantic Web. 1st Semantic Web Working Symposium (SWWS-1), Stanford, CA (2001)
2. MPEG-21, http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm
3. IPROnto, DMAG contribution to MPEG-21 Rights Data Dictionary and Rights Expression Language CfP. http://dmag.upf.es/ontologies/ipronto/mpeg21/m7675(DMAG).zip
4. IMPRIMATUR Project, http://www.imprimatur.net
5. Gallego, I; Delgado, J and Acebron, J.J.: Distributed models for brokerage on electronic commerce. In Trends in distributed systems for electronic commerce, Springer, Germany.1998. ISBN: 3-540-64564-0. 129-140.
6. Gallego, I., Delgado, J. and García, R.: Use of Mobile Agents for IPR Management and Negotiation. In Mobile Agents for Telecommunication Applications, MATA'00. LNCS 1931, ISBN: 3 540 41069 4. Springer, Germany, 2000, pp. 205-213.
7. MPEG-21 Use case scenarios v2.0, N4330, Section 15.3, http://ipsi.fhg.de/delite/Projects/MPEG7/Documents/W4330%20M21%20Use%20Cases.html#_Toc508698290
8. Tous, R. and Delgado J.: Advanced Meta-search in the Web. Elpub2002 (to be published) http://www.tu-chemnitz.de/elpub02
9. Lassila, O., Swick, R.R. (editors): Resource Description Framework (RDF), Model and Syntax Specification. W3C Recommendation 22 February 1999 http://www.w3.org/TR/REC-rdf-syntax
10. Brickley, D. and Guha, R.V. (editors): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft 30 April 2002. http://www.w3.org/TR/rdf-schema
11. FIPA, http://www.fipa.org

12. Eastlake, D.; Reagle, J.; Solo, D. (editors): XML-Signature Syntax and Processing . W3C Recommendation 12 February 2002 . http://www.w3.org/TR/xmldsig-core
13. Delgado, J.; Gallego, I. and Perramon, X.: Broker-based secure negotiation of Intellectual Property Rights. Information Security, 4th International Conference, ISC2001. Springer-Verlag, 2001. Lecture Notes in Computer Science (LNCS), Vol. 2200. ISBN 3-540-42662-0.
14. ICS-FORTH RDFSuite, http://www.ics.forth.gr/proj/isst/RDF
15. DAML-S Coalition: DAML-S: Semantic markup for Web services. In Proc. Semantic Web Working Symposium (SWWS), 411–430, 2001.
16. RDFLogic, http://www.w3.org/2000/10/swap/log
17. RuleML, www.dfki.de/ruleml
18. Darpa Agent Markup Language + Ontology Interchange Language (DAML+OIL), http://www.daml.org
19. Zou, Y.: DAML XSB Rule version 0.3. http://www.cs.umbc.edu/~yzou1/daml/damlxsbrule.pdf
20. JADE-LEAP, http://leap.crm-paris.com/
21. Jess, http://herzberg.ca.sandia.gov/jess
22. DAMLJessKB web page plan.mcs.drexel.edu/projects/legorobots/design/software/ DAMLJessKB
23. Sun, http://www.sun.com
24. Mobile Information Device Profile, http://java.sun.com/products/midp
25. Pjava, Personal Java. http://java.sun.com/products/personaljava/pj-emulation.html
26. Java 2 Platform Standard Edition, http://java.sun.com/j2se
27. W3C Web Services Activity. http://www.w3.org/2002/ws
28. McIlraith, S. and Son, T.C.: Semantic Web services. IEEE Intelligent Systems (Special Issue on the Semantic Web), 16(2):46-53 (March/April 2001)
29. Universal Description Discovery and Integration, http://www.uddi.org
30. Web Services Description Language, www.w3.org/TR/wsdl
31. Simple Object Access Protocol, www.w3.org/TR/SOAP