J-ALINAs: A JADE-based Architecture for Linguistic Agents

Savino Sguera, Armando Stellato, Donato Griesi, Maria Teresa Pazienza
DISP, Università di Roma "Tor Vergata"
s.sguera@ieee.org, {stellato, griesi, pazienza}@info.uniroma2.it

***Abstract*—In this paper we present J-ALINAs, a JADE-based Architecture for LINguistic Agents.**

The purpose of this architecture is to support communication between agents whose beliefs and intentions are driven by different, heterogeneous, knowledge models. This objective, often referred in literature as *semantic coordination*, can be carried on through the identification of specific agent roles and behaviors dedicated to the mediation of agents' knowledge.

In particular, such minimal hypotheses suggest an intelligent exploitation of natural language based technologies (resources and systems) as a necessary choice for capturing those similarities between the different knowledge models of agents trying to communicate, which are not in any way formally ratified.

We aim to provide a flexible framework to be adopted in open multi-agent environments across different scenarios, providing a further abstraction level from the underlying details related to specific semantic coordination approaches; a high-cohesion and low-coupling design, and an agent-interaction protocol make possible for the architecture to face non-*ideal* use cases optimizing the communications among the agents.

We discuss significant design issues, provide a prototypical implementation based on the JADE platform and a case study – MAPLE – integrating an ontology mapping component in the framework, showing flexibility of the architecture in real applications and its independence from any specific mapping algorithm.

Finally we will look at the semantic coordination protocol we designed from a strictly formal perspective, providing a CCS (Milner's *Calculus for Communicating Systems*) description of the protocol itself.

***Index Terms*—Semantic coordination, ontology mapping, cooperating systems, intelligent agents.**

I. Introduction

Enabling communication between heterogeneous semantic peers (agents, semantic web services etc…) is a fundamental issue for the developing of the Semantic Web, especially in relation to the high levels of heterogeneity, evolution, distribution and autonomy of information which already characterize the Web as we know it now.

For the agents to be able to carry out the tasks they were designed for, they must impact the *communication barrier* between themselves: they will have to make their services available to the community, and recognize those of other *actors* in the community (to whichever extent it is considered, up to the Web as a whole); recognize, interpret and respond properly to communicative acts initiated by other agents and *understand* messages' content.

Although an important effort has been done by FIPA (Foundation for Intelligent Physical Agents) towards a standardization of agent platforms and agent message transport, too many *dimensions of variation* still are to be taken into account.

First of all, while adherence to the FIPA message protocol grants agents with the ability to establish a communication, distinguish between requests and responses, and bind them to a given subject of conversation, nothing is said about the real content of these messages, thus making impossible for agents belonging to different communities to actually communicate. We still have to cope with the heterogeneities of agent societies and, mostly, with the diversities in agent frameworks design which are often developed for ad-hoc environments and applications. All these aspects represent a threat against *large-scale* interaction among different multi-agent systems.

If peers were able to autonomously discovery each other's services and to identify the specific problems which impede their communication, it would be much easier to devise solutions for supporting their communication, having a solid base to address the *impedance mismatch* among agents' different knowledge representations from a shared starting point.

II. Our Approach

In a scenario like the one depicted above, it appears evident that agents cannot rely on any shared form of understanding, their inner knowledge, as well as the functionalities they expose, being expressed and modeled upon ruling principles which are not known a-priori.

Our approach aims to:

- identify the fundamental *actors* and *roles* involved in the semantic coordination activity
- define agent-based mediation paradigms,
- *design* an agent-interaction protocol to support semantic coordination in a *flexible* fashion, and
- *model* the semantic coordination process giving the designed framework a high abstraction level from the

specific instance of the process itself.

Following these guidelines, we designed and developed J-ALINAs[1], a (JADE based) Architecture for LINguistic Agents: an agent framework for supporting semantic coordination between heterogeneous semantic peers. Main feature of this architecture is the abstraction from the *way* actually semantic coordination is carried out by a specific instance of an agent system, still considering it the main shared goal for the community. While abstraction from the techniques is maintained, on the contrary, it is important to identify the different kind of resources which may play a role in the process, in order to make them available when their contributions is required. In particular, we consider as necessary an intelligent exploitation of natural language (from which the name J-ALINAs arises) resources and processors, as language is the sole form of shared knowledge which is inherently adopted when expressing (through label descriptors, concept documentation etc..) the formal content of knowledge models and resources.

Moreover, we achieved a clear distinction between decision-making and service-providing competences, realizing a crucial separation between specific problem-solving skills and *strategic* knowledge about their composition to address a complex target, shared among the system's agents.

This allows to analyze new problems simply rearranging existing elements, and to improve or add functionalities without necessarily modifying the entire system.

III. Previous Works

A. State-of-the-art

A lot of work has been done by researchers towards flexible architectures to support semantic coordination in open multi-agent systems.

In [5] a three-layered peer-to-peer approach to ontology integration in a multi-agent system is described, to manage and deploy ontologies in a broad range of dynamic environments.

Within a multi-agent system however each agent expresses its own conceptualization, and in open systems interoperation between agents has to take into account their heterogeneous nature and background, that will likely take them not to fully understand each other, because of semantic misalignments which can easily arise.

A major requirement for agents interoperability [3] is for the semantic integration to be dynamic, that is, computed on-the-fly and not on the basis of pre-engineered mapping documents (which, even if considerable, may not always be available for any two given knowledge models). Agents shall have to be *self-describing*, and be able to characterize themselves, thus putting other agents in the condition of identifying the ones they need to cooperate with, in order to successfully communicate and exchange information towards a shared goal – that is in our case the agents' ontologies mediation.

To put it in Burstein's and Uschold's words [3], "*every agent must, in effect, wear its description 'on its sleeve'*".

Moreover, some interesting links between multi-agent systems and grid computing are given in [7], the formers complementing the latter for efficient services management without a-priori agreements, standing that multi-agent systems are groups of agents interacting and autonomously coordinating to satisfy a set of shared goals.

B. ALINAs: an open interacting-agent architecture

ALINAs [8], an Architecture for LINguistic Agents, developed by the Artificial Intelligence group of Tor Vergata, and former incarnation of the architecture which is presented in this paper, provided a set of three different classes of intelligent agents dedicated to supporting linguistic communication, each of which exhibits specific features related to the particular task it has been designed for:

- **resource agents**, which represent the beneficiaries of the communicative process, bearing some form of knowledge which need to be mediated against the one of other resource agents in the community
- **service agents**, providing support functionalities and holding responsibilities for some complex tasks occurring in the process
- **control agents**, having thorough knowledge of the problem to solve, control functionalities and decisional power in the agent society

This classification provides enough abstraction to ensure an incremental approach to systems development, whereas developers can focus on informative sources and resource agents and start to use available information before they design more advanced components in the system.

C. Linguistic Watermark and Semantic Coordination

As stated in [10], in order to make ontologies more prone to be mapped in distributed contexts, we believe it is necessary to revise the ontology development process to include, as a necessary part of this activity, the *enrichment* of ontology content with proper *lexical* expressions in natural language, such as synonyms and free natural language documentation, possibly in different languages.

Linguistic enrichment of ontologies (this is the name we gave to the process here described) requires a proper exploitation of several linguistic resources, which, due to the lack of defined standards for representation of linguistic knowledge, often differentiate upon many aspects, like structure, semantics, granularity of their content and representation. This strong heterogeneity led us to the development of the Linguistic Watermark [1], which is both a package providing generic abstract classes and interfaces for allowing uniform access to different linguistic resources, as well as a set of descriptors for identifying the characteristics of the accessed resources. These descriptors are important in our environment, as they allow agents to choose the linguistic resources (and thus contact the agents which grant access to their content) which are more appropriate for supporting a given communication.

Following the same intuition, it is also important, should an

[1] http://ai-nlp.info.uniroma2.it/software/J-ALINAs

ontology have been already linguistically enriched, to know in advance to which extent and through which modalities this enrichment process has been conducted. In our framework, this information is represented by the Ontological Linguistic Watermark [10], a collection of meta-data descriptors (see Figure 1) which expresses information about the (natural) language(s) adopted in describing ontology contents, and (eventually) about the linguistic resources which have been used to do that. These metadata play a central role in the semantic coordination phase we will describe extensively in section V.

```
<Linguistic Watermark> ::= { ontology enriched by: <linguistic_resource> }

<linguistic_resource> ::=
                <linguisticResourceURI>,
                [ semantic_enrichment | linguistic_enrichment, <language> ]
                <enrichment_modality>
                <coverage>

<enrichment_modality> ::= [ supervised_enrichment | automatic_enrichment ]

<coverage> ::=  <conceptual_coverage>, <linguistic_coverage>
```

Figure 1: Specification of the Ontological Linguistic Watermark

D. MAPLE

MAPLE is a plug-in for the Ontology Editor Protégé [4], developed at the Artificial Intelligence research group of University of Rome, Tor Vergata, aimed to integrate ontology mapping facilities into the Protégé [4] Ontology Editing Suite.

It relies upon external linguistic resources – compliant to the Linguistic Watermark package – to obtain further information useful to find semantic correspondences between ontologies. The mapping process, before a deep inspection of the semantic characteristics of the two ontologies, preliminarily requires, at the linguistic level, to search for alternate expressions (synonyms) and/or glosses for the labels which describe ontology content, and also to identify proper translations in the more complex situation occurring when ontologies are labeled using different (natural) languages.

IV. J-ALINAS' ARCHITECTURE

A. The big picture

The generic meaning negotiation process between semantic peers usually has to start with a first hand-shake, in which information about the peers' respective knowledge is exchanged (in the form of ontology namespaces, as an example). Should the two agents find that they are not able to communicate due to any form of incompatibility between their form of knowledge, they will first try to coordinate in order to invoke the figures which can support their communication.

Suppose an agent wants to query a search engine, to obtain a link to a remote document; so far the scenario is particularly known (Figure 2). We then introduce this situation in the Semantic Web context, whereas documents will be expressed *against* a formal description of the content they describe: an ontology.

The agent, to effectively understand the document's content, will likely have the need to find mappings between its set of beliefs and conceptualization (its ontology), and the ontology the document is referred from.

To carry out the meaning negotiation and reach a knowledge model which is acceptable for both peers, the agent will request mapping service to a dedicated agent; the latter will take control of the whole negotiation process and will eventually need to request other agents' support or information regarding involved ontologies.

However, it is particularly interesting, again, to notice how this scenario is *independent* from the mapping techniques the dedicated coordinating agent will implement, whether they be based upon finding lexical anchors upon mapping documents or aimed to *spot* schema-level similarities, or to perform a combination of both.

The JADE[2] platform has been adopted to provide a prototypical implementation of the framework, in order to obtain a fully platform-independent prototype. JADE actually simplifies the development of agent platforms through a middleware complying to FIPA standards, implementing some of the required agents for the platform to be FIPA-compliant, and allowing agents to be movable from one machine to another.

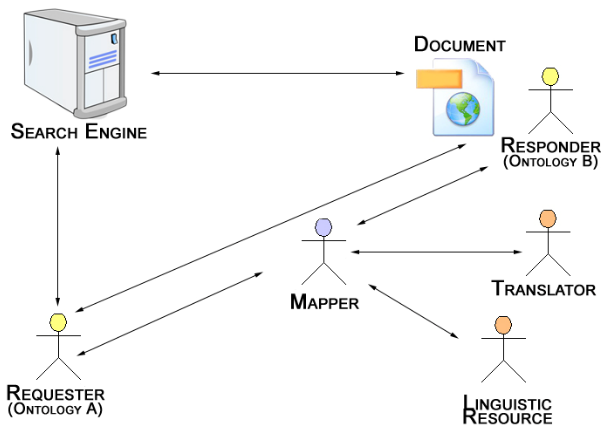

Figure 2: A generic application scenario

B. Agents and roles

J-ALINAs maintains much the same classification of agent roles which has been presented in section III.B, and more in details in [8] and [9]. Yet this approach is not to be intended as a rigid one: indeed the boundary between – for instance – service and resource agents will sometimes be not so clear, and will be possible for an agent to behave on both sides.

For this reason we believe it will be – in certain circumstances – more correct referring to resource, service or control *roles* rather than *agents*, since an agent could embody more than one role in the mediation process.

In other words it will not necessarily exist a *straight* 1-to-1 relation between an agent and the role it assumes in the

[2] http://jade.tilab.com

society, even though in the following paragraphs we will suppose it to exist, merely to have a distinct view of the identified competences. Here follows a more detailed description of the roles we have mentioned so far.

Linguistic agent

Linguistic agents encapsulate one or more linguistic resources, and provide an interface to access their content, through different kind of services.

The agent registers itself at a *Directory Facilitator* (DF) living in the agent platform, and publishes its services and the resources' URIs.

Ontological agent

An ontological agent is – together with the mapper agent – one of the fundamental actors of the process: usually it is such an agent to start the semantic coordination procedure and to request a conceptual mediation against another agent's ontology. He shall then assume the **requester** role (with this name we will refer to the ontological agent requesting the meaning negotiation), and choose the **responder** (the ontological agent holding the *destination* ontology); after that *he* will query the DF to search agents providing mediation services. We will discuss the **handshaking** phase in details in section V.

For the *semantic coordination* to take part in an effective way, each of the ontological agents willing to communicate publish a *fingerprint* of the conceptualizations they encapsulate (the Ontological Linguistic Watermark, OLW), containing information about the linguistic enrichment processes (as described in section III.C) which contributed to describe their knowledge content. As the ontology may have not been subject to any linguistic enrichment process at all, the OLW may even be reduced to the sole information about the idiom used to represent its concept identifiers. By comparing ontologies' watermarks, the mapper agent will then be able to easily decide which supporting agents are likely to be contacted, and negotiate the (natural) language(s) to be used throughout the whole mapping process.

Mapper agent

The only agent in the society to have the needed intelligence and a thorough knowledge of the problem, of the actors and of the methodologies to be applied to carry on a meaning negotiation activity, is the mapper agent.

We first need to distinguish between two different kinds of mapper agents: those which provide semantic coordination facilities by inspecting available ontology mapping documents, that is, resources available on the web which state conceptual correspondences between ontology resources, and those which offer the same service by computing those correspondences on-the-fly.

The first agent will mostly carry on basic look-up over available mapping documents (even more than one, as agent ontologies may have been built compositionally from several available smaller ones), supported by inferential abilities to entail new mappings other than those which are explicitly declared.

For those cases where mapping documents are not available (or they do not cover completely the addressed conceptualizations) – presumably not a *small* fraction of *real world* situations – the second kind of mapping agent is invoked to try to individuate and establish possible semantic similarities by inspecting the structure of the two ontologies and by exploiting information provided by linguistic agents.

C. The agents' behaviors

We designed a set of JADE behaviors, realizing the system's reactive layers hierarchy. In the following paragraphs we will describe the most significant ones, *moving* our perspective from one agent to another.

1) Linguistic support to ontology mapping

The behavior encapsulating access to linguistic resources and providing linguistic support to the ontology mapping process is a cyclic one, enabling the linguistic agent to react to mapper agent's stimuli. In other words it models a simple reactive behavior, implementing a reactive agent paradigm.

A linguistic agent in the *setup* procedure will add an instance of such behavior to its execution queue; it will eventually reply to queries originated by the mapper agent with semantic anchors for the particular term, or with a *NOT_UNDERSTOOD* FIPA message performative in such cases where the query is malformed (or, more simply, out of its comprehension).

Queries are matched by the agent against an application-defined message template, providing developers with high freedom of choice with respect to the type and number of linguistic services the agent publishes in the system.

The *linguistic-querying* task is delegated to a specific behavior residing in the mapper agent (Figure 3), which is responsible for persisting anchors upon the agent itself; however the concept of *persisting* anchors is absolutely abstract for the framework, and it is responsibility of the application developer to implement the *persist* and *retrieval* functions.

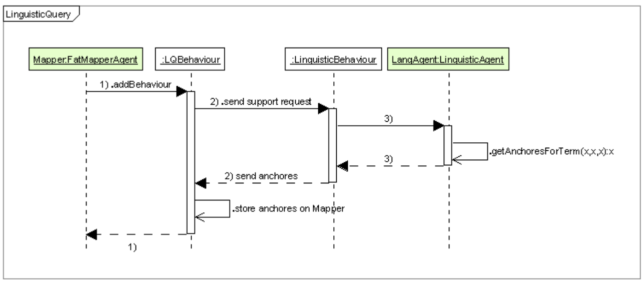

Figure 3: Interaction between mapper agent and linguistic resources

2) The twofold ontological agent behaviour

The provided ontological agent paradigm has been designed to act indifferently as *requester* or *responder* in the mapping process. We actually reckon this has positive implications whether towards bidirectional knowledge mediation between agents, and the integration of the application-specific ontological agent into the surrounding environment, reducing the number of different *actors* for the developers to take into account.

If the agent is required to act as *requester* in the process it will start discovering the surrounding environment, contacting agents and *handshaking* them.

Otherwise, if the stimulus is a *PROPOSE* FIPA message performative, the agent will eventually accept to participate in a mediation and wait for a mapper agent to ask for its ontological linguistic watermark or information about its ontology's concepts and relations among them.

3) The ontological agent's introspection

Concurrently, the ontological agent runs an instance of a *simple reflex* behaviour aimed to answer specific queries about ontology's instances, properties or relations among concepts. This behaviour can be exploited by developers to fit messages exchange among agents to support the specific mapping algorithms, and does not affect the handshaking procedure at all.

4) The generic ontology mapping process' instance

Modules providing ontology mediation based upon external linguistic resources have been designed to give the handshaking procedure a *well-known* form, and at the same time to give developers using the framework the chance to easily *extend* the system with custom behaviours encapsulating application-specific mapping algorithms.

Indeed, after the *handshaking* phase took place the mapper agent shall own all the elements needed to conduct a *generic* ontology mediation, and to effectively decide which other agents in the society are likely to be involved in the process; in other words we provide developers embracing J-ALINAs architecture a solid and shared base to design their application on, building a further abstraction level from the underlying environment.

The abstraction itself comes to evidence even from a strictly *sequential* perspective: because every mapping process starts with a communication among agents and the system's facilitator whereas the agents discovery the surrounding environment, we reckon it is convenient to let developers the chance not to care – to some extent – about the *coordination* and *synchronization* layer, focusing on their system's peculiar aspects.

V. Semantic Coordination: The Agent-Interaction Protocol

The interaction protocol for the semantic coordination is intended to be the whole project's *keystone*, designed to adapt itself to different situations, minimizing the number of messages the agents must exchange – as it is normal to expect in a distributed environment – to achieve their communication's main purpose: the mapping of agents' ontologies (or, more specifically, of the concepts they need to express in their communication).

Throughout the protocol requisite elicitation and analysis phases, the chance for an agent to *embody* more than one role in the mediation process initially emerged as a problem: indeed, this implies the agent to be *aware* of its manifold functions while interacting, discovering and choosing which other agent to query. For instance, an agent implementing both the mapper and the ontological paradigm could prefer to rely upon its own mapping *skills* instead of contacting a third-party.

This brought the need for the protocol to adapt itself to variations in the system's *topology* (Figure 4) and in the message flow among agents, and at the same time to ensure developers freedom of choice in deploying functionalities on application-specific agents.

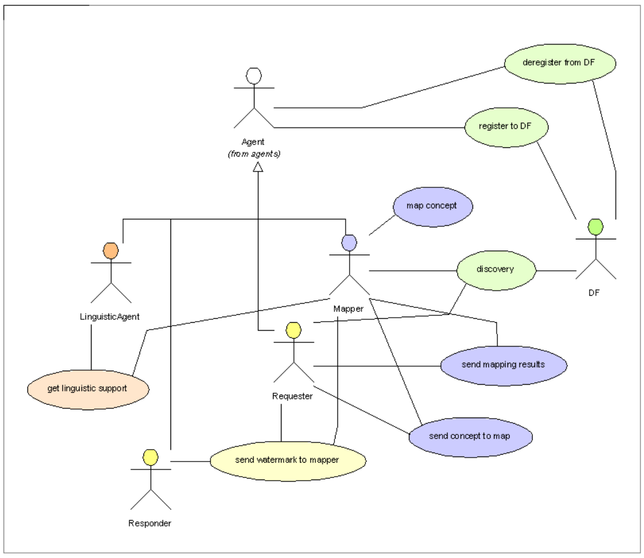

Figure 4: J-ALINAs use case diagram

After the setup phase, the agents are in a stand-by state. Whenever the *requester* receives the stimulus to initiate the process, an *handshaking* activity (Figure 5) introduces the communication. The *DF* will be queried to discover which other ontological agents are living on the platform. The *requester* agent will then elaborate the search results, and choose the passive agent (*responder*) in respect to fully application-defined criteria.

The mapping proposal is sent to the chosen *responder* agent, who will eventually accept the proposal, notifying the *requester* with an *ACCEPT_PROPOSAL* FIPA message performative including its ontology URI.

The *requester* will then query the *DF* asking for the presence of formal-model based mapper agents providing mappings between its ontology and the *requester's* one.

Whether such an agent exists in the platform, the *handshaking* phase stops here, and comes the time for the *requester* to start querying the mapper with concept-mapping requests.

In the other case, another query is submitted to the *DF*, this time looking for the existence of a mapper agent based upon external linguistic resources exploitation. If the query is successful, the *requester* shall choose which mapper agent to contact and send out a *PROPOSE* message performative, waiting for the eventual acceptance from the counterpart. The proposal message will include the agent's ontological linguistic watermark and the *responder* unique identifier.

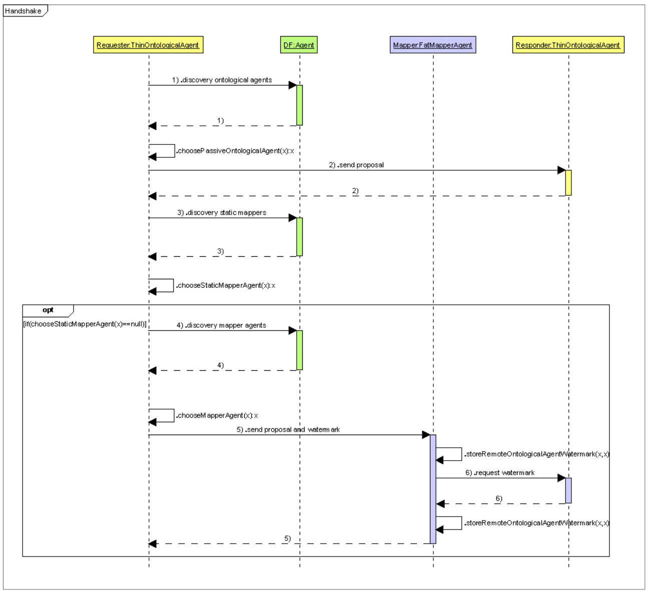

Figure 5: Handshaking - sequence diagram

The semantic coordination is now complete: the mapper agent has all the information needed to negotiate the natural language to use throughout the mediation, and choose which other agents to contact; the actual ontology mapping process can now start. Whilst no other agent was holding control of each others' actions and of message flow in the system so far, now the decisional power goes completely to the mapper agent.

The adoption of such a protocol does not violate the dynamics of the process and the heterogeneous nature of actors which are typically associated to the idea of Semantic Coordination. On the mere perspective of ontological agents (which could be considered as the *end users* of Semantic Coordination), they just need to be *aware* of the existence of Mapping Agents willing to help them in the process of communicating with other agents based on different kind of knowledge, and implement (as they are expected to do in whatsoever scenario) basic speech acts for requesting their help. Knowledge of an ontological agent's own linguistic expressivity is also requested in our paradigm, but this in line with recent trends in the Semantic Web area, where ontologies need to be expressed in a linguistically motivated fashion [2,7], possibly considering integration with existing linguistic resources [1]. The core of the mapping process is then delegated to service agents and resource agents, thus not requiring any strict protocol to be followed by ontological agents.

VI. A Case Study: MAPLE

As anticipated in III.D, MAPLE is a Protégé plug-in integrating ontology mapping functionalities in the Protégé ontology editing tool. MAPLE also provides ontology mapping as a standalone process, though it is not conceived for a distributed environment.

We succeeded in abstracting MAPLE's algorithms and their relations with linguistic resources from its original scope, and provided an implementation for a mapping agent which *fits* in a distributed environment according to the described framework.

From MAPLE's algorithms' analysis what soon emerged is a linear relation between the *cardinality* of the *target* ontology and the number of *queries* the mapper agent is supposed to submit to linguistic resources (especially for the resource which is expressed in the natural language adopted for the communication). Standing on MAPLE mapping algorithms, not all the queries can be estimated in advance and sent as a unique request to a linguistic agent, thus requiring an heavy communication load between the mapper and the linguistic agents. This makes unacceptable the approach with autonomous linguistic agents to hold information sources, because the *size* of the involved ontologies is obviously not estimable in advance.

So we looked for a trade-off between the mapping process' distribution and the communicational complexity among agents: this resulted in one single agent implementing both the mapper and the linguistic agent paradigms. This way we kept constant and independent from ontologies' size the number of messages to be exchanged in the system, encapsulating in a single agent – thus in a single physical host – mapping algorithms and linguistic resources accesses.

The eventual need for a translation phase can easily be delegated to an external agent: the label(s) associated to the concept to map can be sent to a linguistic agent providing the translation services which will then send back to the mapped the translated terms; the mapper will then try to individuate the right translations among those provide by the translator, and then start to work in the language it is specialized in. This particular framework individuates a specific figure for mediation, given by a mapper agent which is particularly proficient in one (or more) idiom and which must be contacted only when its idiom pertains to at least one of the communicating ontological agents' OLWs.

VII. Conclusions

One of the most interesting perspective to look at JADE – and obviously at JADE-based systems – is the possibility to easily design – through extremely interesting extensions such as JADE-LEAP (Light Extensible Agent Platform) – agents living in mobile platform and devices.

Looking at the Semantic Web evolution together with the proliferation of wireless devices – PDAs and mobile phones – it becomes quite clear what the web is likely to become in the next years: a *dense* and highly dynamic network, populated by a multitude of *nomadic* elaboration nodes able to *understand* information sense, communicating in a more effective and *intelligent* way.

We believe it is natural – to some extent – to look at these *nodes* as autonomous entities communicating in a peer-to-peer fashion, *living* in an heterogeneous environment, providing and asking services one to each other; that is, actually, as agents.

As we already discussed, semantic coordination will be an essential and binding part of this communicative process: when most of the information will be expressed with respect to these specifications; knowledge mediation will be a very frequent *runtime* process, providing the base for mutual comprehension between systems.

Also, we believe a distributed approach has to be considered fundamental. This is one of the most interesting perspective to look at our work, which is actually providing a *flexible* framework for multi-agent systems development in response to the different needs coming from several backgrounds and practical applications.

We took into account the communicational effectiveness and efficiency, providing the framework with enough *granularity* to let developers easily control and keep *constant* the number of messages exchanged by the agents living in the particular instance of the system, leaving open the road to developments in scenarios with very specific constraints like those expressed in the *mobile computing* area.

APPENDIX

The handshaking procedure: a formal review

In this paragraph we will give a – simple – formal description of the handshaking phase and the coordination protocol we designed, through Robin Milner's *Calculus for Communicating Systems* (CCS), [6]. More specifically, what we will do is looking at each agent of the society as an independent process, and model its behaviour in terms of others'.

We first define the generic *Directory Facilitator* as a recursive process:

$$\boldsymbol{DF = (search + searchFederated) . \overline{result} . DF}$$

We then define the twofold ontological agent behaviour as follows:

$$OA \xrightarrow{Req} OA_1$$
$$OA \xrightarrow{Prop} OA_2$$

where *Req* stands for the receiving of a REQUEST FIPA performative, and *Prop* for the receiving of a PROPOSAL FIPA performative, and specify that:

$$\boldsymbol{OA_1 = (\overline{search} .result.chooseP. \overline{proposeP} . \overline{search} .result). (chooseFM + \overline{search} . chooseM . \overline{proposeWM})}$$

where *chooseP* and *proposeP* indicate respectively the actions of choosing the passive ontological agent and notifying it with the proposal to collaborate in the mediation process, and *chooseFM* and *chooseM* indicate the action of choosing the Formal Mapper agent, where available, and

$$\boldsymbol{OA_2 = proposeP . \overline{watermark} .}$$

The mapper agent will behave as follows:

$$\boldsymbol{MA = proposeWM . storeWM . watermark . storeWM}$$

where *WM* stands for Watermark

Now we can define the whole *handshaking* process as processes above running concurrently and stimulating each other, limiting each process' interface to allowed communication channels:

$$\boldsymbol{Handshake = (OA_1 \mid DF \mid MA \mid OA_2) \backslash chooseP \backslash chooseFM \backslash chooseM \backslash storeWM. \backslash searchFederated}$$

REFERENCES

[1] Benjamins, V. R.; Contreras, J.; Corcho, O.; and Gómez-Pérez, A. 2002. Six Challenges for the Semantic Web. In Proceedings of SemWeb@KR2002 Workshop, 19-20 April 2002, Toulose.

[2] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, P. Cimiano. LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies In Proceedings of the OntoLex Workshop at LREC, pp. 28-32. May 2006.

[3] M. Burstein, M. Uschold: "Infrastructure for Semantic Interoperability and Integration: breakout discussion summary", Dagstuhl Seminar Proceedings: Semantic Interoperability and Integration 2005

[4] Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubézy, M., Eriksson, H., Noy, N., and Tu, S.: The evolution of Protégé-2000: An environment for knowledge-based systems development. International Journal of Human-Computer Studies, 58(1):89–123, 2003.

[5] L. Li, B. Wu, Y. Yang: "Semantic mapping with multi-agent systems", 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)

[6] R. Milner: "Communication and Concurrency", Prentice-Hall International, Englewood Cliffs, 1989.

[7] A. Oltramari, C. Huang, A. Lenci, P. Buitelaar, C. Fellbaum. OntoLex2006: Second Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies, held jointly with LREC2006, Magazzini del Cotone Conference Center, Genoa, Italy

[8] M. T. Pazienza, A. Stellato, M. Vindigni: "ALINAs: un'architettura multi-layer ad agenti per il supporto alla comunicazione linguistica", WOA2002, Milano (Italy), 18 November 2002

[9] M. T. Pazienza, M. Vindigni: "Agent based ontological mediation in IE systems", LNAI 2700, Summer Convention on Information Extraction 2002, Information Extraction in the Web Era, pp. 92-128

[10] M. T. Pazienza, A. Stellato: "Linguistically motivated Ontology Mapping for the Semantic Web", SWAP 2005, the 2nd Italian Semantic Web Workshop Trento, Italy, December 14-16, 2005

[11] M. T. Pazienza, A. Stellato An Environment for Semi-automatic Annotation of Ontological Knowledge with Linguistic Content 3rd European Semantic Web Conference (ESWC 2006) Budva, Montenegro, June 11-14, 2006