Semantic Resource Management in MAS

Nicola Cannata, Flavio Corradini, Francesca Piersigilli, Emanuela Merelli and Leonardo Vito
Dipartimento di Matematica e Informatica
Università di Camerino, Via Madonna delle Carceri 9, 62032 Camerino (MC), Italy
Email: name.surename@unicam.it

***Abstract*—In highly dynamic environments like academy and industry it is becoming essential the need of efficient systems for resources organization and discovery. In this paper we describe a semantic resources manager, called Resourceome. This system allows both to discover and organize resources for agents' goals achievement. The ontological descriptions of resources and of domains allow to contextualize a resource instance in its domain through a *concern* relation. The proposed model supports the navigation from domain to resource concepts and vice versa. Resourceome represents our proposal for describing the particular vision of the world perceived by multi-agent systems.**

I. Introduction

The wide use of distributed systems led to the design and implementation of middlewares. Corba, RMI, Web Services, and FIPA are the most important standard specifications, which gave rise to successful middlewares in both business and academic environment. Even if actual middlewares allow developers of distributed applications to overcome the interconnection and integration troubles, it still remains present the need to have suitable support to organize and discover resources to fully support the systems interoperability. For *resources* we mean web services, persons, tools, databases, files and others available either on the web or locally. Concerning pharmaceutical industry and bioinformatics research we are witnessing a growing number of published resources, if properly organized could be the basic knowledge of artificial systems in life science [1].

In any distributed system, likewise an agents community, the organization of the resources plays a very important role. In fact, it is often difficult for a software agent to look for the right resource in an unexplored open environment [2]. Agents generally do neither know what kind of resources are available, nor if a certain resource is still existing in their environments; even playing a specific role. An organization of contextualized resources in their domains can help their discovery (in particular at run-time) by a resource manager. This could also replaced a resource with another one when the original one is not available or when an equivalent or better one is found. The dynamic and distributed scenario is the natural environment for multi-agent systems (MAS). In such context, where the aggregation of new communities of agents is possible, the semantic discovery of resources would be very useful.

To this end, we propose a model for a semantic resources manager, called Resourceome. System that allows both to discover and organize resources contextualized in their domains. Resources are described by suitable ontology whose instance resources are related to their domain concepts by a specific relation, that in the proposed model, is called *concern*. The proposed model supports the navigation from domain to resource concepts and vice versa.

Resourceome model differs from existing models, as OWL-S and WSDL-S, in two main features: 1) Resourceome allows the description of any kind of resources, e.g. web service and ; 2) it allows to add new resources and to contextualize them on appropriate domains.

Resourceome with its model, thus, represents our proposal for describing the particular vision of the world perceived by multi-agent systems.

The rest of this papers is organized as follows. Section II introduces the concept of Resourceome. In Section III the Resourceome management system is described. Section IV shows how the Resourceome can help software agents to discover resources. A case study of resources discovery with Resourceome is presented in Section V.

II. What is a Resourceome?

In this section we describe the basic model of Resourceome. The basic idea behind Resourceome is that of a formal, machine understandable description of resources through two ontologies: the domain ontology and the resources ontology [3].

The main purpose of the resources ontology is to organize the concepts, properties and relationships through which a resource can be classified. Whereas the domain ontology organizes the topics the users are interested in. We can foresee general purpose resourceomes to organize resources in industry, computer science, bioinformatics or something else. For example, industrial topics could be represented by concepts like *electricity*, *energetic saving*, *electronics* and *staff management*. The domain ontology defines the resources context, allowing humans and software agents to easily understand what a resource is about.

Thus, a resource context is also given in terms of concepts of the domain: in our representation resources are linked to the concepts of the domain ontology through a specific relation called *concern*. Figure 1 describes the Resourceome model in terms of concepts (ovals) and *is-a* relationships (arrows) between concepts, and *concern* relationships between ontologies (dark arrows). Resourceome can manage any ontology provided there exists at least one *concern* relation. In the sequel of the paper, for sake of simplicity, we have only used, as examples, ontologies with *is-a* relation.

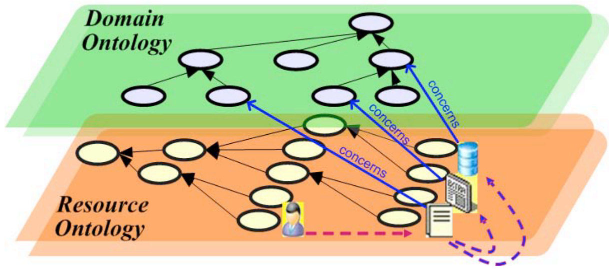

Fig. 1. The Resourceome model

To define the *concern* relation we aimed to use OWL-Full, but actually it doesn't exist a reasoning software able to support complete reasoning for every feature of OWL Full. Consequently, we used OWL-DL, even if its expressiveness was not sufficient - OWL-DL can be used only under certain restrictions, for example, a class cannot be in relation with an instance of class-. To overcome the OWL-DL restrictions we have introduced an *hidden* instance for any ontological concept.

A domain could eventually be specified through more than one ontology, e.g. for interdisciplinary domains. Furthermore and more concretely, resources are univocally defined through their metadata and in particular through their URIs [4], that can be LSIDs [5], DOIs [6], URLs, and so on.

A. An example of Resources Ontology

The notion of resource is fundamental in current networked information systems. The term is used often, specifically in relation to World Wide Web and Semantic Web activity, e.g. in standards such as RDF [7]. This term masks an exceptional amount of ambiguity. Although a stated definition of a resource in the URI RFC [4] exists, it is in many respects vague: "A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings and books in a library can also be considered resources [...] a resource can remain constant even when its content, the entities to which it currently corresponds, changes over time, provided that the conceptual mapping is not changed in the process. [...]". In an industrial scenario, the types of related resources depend on the production domain, even though we can affirm that there are some resources common to every sphere. Just to give an idea, Figure 2 shows some concepts that can represent a fragment of knowledge in an industrial scenario. In particular, *warehouseman*, *firm employee*, *person agent*, *software agent*, *human agent* are concepts defined by the following relationships:

- a Warehouseman is-a Firm Employee
- a Firm Employee is-a Person
- a Software Agent is-a Agent
- a Human Agent is-a Agent

Also a *service agent* [8] can be considered a resource, in fact, in our proposal - as it is described in the sequel - a service agent has the possibility to register itself as a resource in the Resourceome of the running platform by reporting its particular features and concerning one or more domain topics.

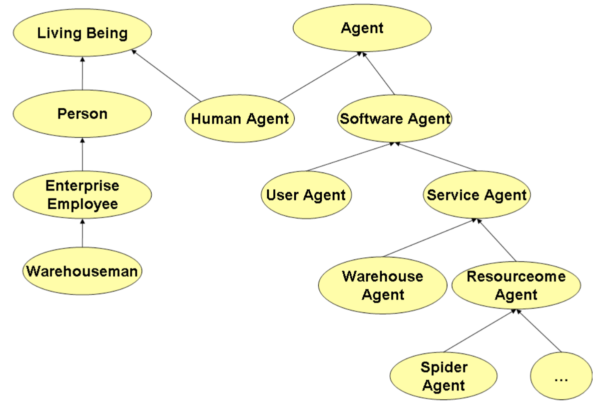

Fig. 2. A fragment of an industrial Resources Ontology

In addition, we can establish more explicit relations between resources like: *uses*, *describes*, *is author of*, *is executed by*, and so forth. And concepts can have many attributes such as *name*, *description*, *URI*, etc.. These relations and attributes allow software agents to obtain information and knowledge about a certain resource, so that they can choose between apparently equivalent resources.

We have developed some examples of resources ontology, however being Resourceome a customizable system, resource ontologies can be built at will.

B. Resources instances

Instances of resources can be considered as an *instances ontology*. Although this component extends the resources ontology, it is more appealing to see it as an independent data set. In the present prototype, it is composed of distributed OWL-DL [9] files referring to a specific resources ontology and concerning a particular domain ontology. The instances files refer to both the resource and domain ontologies by importing them through the definition of their *namespaces*. Individuals can have limited life span, or can be irregularly available. Therefore there is the problem of how to manage no more or temporary not reachable resources. As we will deep in Section III, this task is performed by a pool of specific software agents that periodically check for resources and their status. When a not reachable anymore resource is discovered, it is marked in a different way, in order to allow the user to be aware of it. The ontologies grow receiving concepts, relationships and instances in two manners: they can be added manually by users or automatically by the software agents.

Resources could be available through the web or could represent local resources in a remote machine.

III. AGENTS MAKE "ALIVE" THE RESOURCEOME

The management of the Resourceome is performed by a multi-agent system whose organization allows to carry out also the interactions with the final users.

Software agents are autonomous computational entities operating in an open dynamic environment [10]. Agents generally interact with each other, also collaborating to achieve common goals. Agents can be seen as a design metaphor for constructing complex systems around autonomous, communicating entities [11]. In this context the multi-agent system allows to consider the Resourceome like a living organism that is composed of a static component formally described (i.e. the ontology) and a set of proactive components that maintain updated the ontology (i.e. the multi-agent system). In order to achieve this goal, we introduce the following specialized agents:

- Query/Answer agent: its role is that of taking any user query, and translate it in SPARQL [12]- the language recommended by the W3C and adopted in this solution -. After having queried the ontologies, the results are translated back in the user language.
- Spider agent: its role is that of surfing Internet to find new resources (and also new concepts) concerning a particular domain of interest. The research is based on some parameters such as URLs and keywords. The outputs provided by the Spider Agent are new instances of resources and eventually new concepts and relations to add to the Resourceome.
- Parser agent: its role is that of parsing a flat file prepared to include a set of resources in the Resourceome. The file, e.g. an XML file, is created by users to index the proper user resources (more details are provided in the Section III-A).
- Text mining agent: its role is that of automatically "annotate" the resources described in a text document. It looks for relations between a certain text document and a set of given concepts and resources by following these steps:
    1) it tries to find the best fitting concept belonging to a resources ontology for classifying the resource described by the document;
    2) it tries to find the best fitting topics belonging to a domain ontology for providing semantics to the resource described by the document;
    3) it tries to enrich the semantics of the described resources adding their relationships with other resources. In particular if the resource to be annotated is an article then an instance of the relationship *cites* can be created for every other literature resource cited in the text.
- Matcher agent: it concerns the matching between the new knowledge eventually found by the spider or parser agent, and the actual Resourceome content.
- Session agent: it is the sessions responsible. In fact, more than one session can be opened by users to access the Resourceome, or by the agents that manage the Resourceome, for read/write operations.
- Monitor agent: it tracks the resources signaling if a resource is currently reachable or not. When no longer available after a reasonable observation time, it can be considered definitively unreachable. The use of URN (like LSID and DOI) instead of URL and the resources monitoring, partially solves the "404 not found" problem [13].

We are aware that many of the proposed agents' roles are still open issues. Nevertheless, we developed those basic features already proposed in literature.

A. How to "feed" Resourceome with new resources

Excluding the monitor and query/answer agents, all the others collaborate, by interacting with each other, to automatically add knowledge to the Resourceome. We aim to formalize, in the next future, the communication protocol by using a MAS methodology.

The instances of the resources ontology could be enriched by the spider agent "walking" across the Internet. It should try to individuate the metadata describing interesting resources, adding this information to the current instances "ontology". The task just described is rather complicated. It is performed by a pool of software agents entrusted - by spider agents - to continuously effect searches on Internet, and look for new available interesting resources. The basic knowledge of the spider agent consists on a set of URLs - that acts as the spider starting points - and a list of keywords - concepts of the domain and resources ontologies -.

When a spider finds a new resource to be added, its metadata are converted in OWL-DL, if specified in a different language. Then the matcher agent imports them into the ontology files. The metadata are connected to the suitable concepts, if they are available.

If the right concepts to identify the resource and/or its domain are not yet provided, they are automatically added by the matcher agent supported by the new resource metadata. The metadata and a vocabulary with the use of a similarity algorithms, help the matcher to reach its goal. When the Resourceome does not contain adequate concepts to describe the new resource, new ones can be added from the remote ontology. Also relationships and properties are imported, if possible.

We have also implemented another kind of search based on a particular flat file that Resourceome users should prepare and publish. This file contains the metadata of the users' resources, which must be indexed by the Resourceome. In this case the work carried on by the Parser Agent is indispensable. The flat file should be built in a standard format in order to let the parser identify the elements of the resource description.

IV. THE RESOURCEOME ROLE IN A MAS

Considering multi-agent systems as virtual extension of the human reality, the agent society, at least partially, should replicate some of its specific aspects [14]. A software agent is basically characterized by the goals it must achieve, by the roles it can play in its existence, by its attitudes and by the contexts in which it lives. A context defines also the conditions by which an agent plays a role and has some goals [15]. A context can change during an amount of time and it is very important that an agent has the ability of understanding its context and how it evolves.

The Resourceome allows agents to understand part of the environmental context. In particular the use of the resources, which of course characterize the context. When an agent realizes to be in a certain context, where it can use some resources, it can decide to acquire a new role. The Resourceome logically represents a knowledge-base accepted by the agents community which shares it. It represents also a system that allows every agent to share its knowledge, in order to enrich the intellectual capital of the organization.

There are some objects, such as resources, that have features dependent from observers. In Searle [16] every institutional fact is defined by constitutive rules, like *X counts as Y in context C*. We think that the Resourceome is able to provide this kind of characterization. In fact a resource identifies a certain entity, i.e a resource instance for a precise context of use bind to a specific domain. When it is communally recognized that a resource *X identifies Y in a context C*, the resource *X* assumes a particular *status*. The status indicates the functionality that the resource *X* can play in the context *C*.

Thanks to this characterization also agents (which in turn represent resources) can be contextualized and semantically discovered through the combination of the Resourceome and of the Directory Facilitator.

A. The broker role of the Resourceome

The Resourceome can be considered a data structure representing formally part of shared knowledge among an agents community. The pool of agents for the Resourceome management, described in Section III, can be seen as an organization whose members have been created to hold specific roles and to interact with each other by respecting the organization laws. Indeed not all agents must necessarily have the ability to directly navigate the ontologies of the Resourceome to gain knowledge about context resources. Any agent can be helped by specialized agents which are the promoters of its requests execution. These agents specialized to query the Resourceome give to the Resourceome system the role of resource broker.

In such a view, the system can find one or more kinds of resources concerning a particular domain, or alternatively, from a given domain concept it retrieves all resources concerning it. It follows a list of some search examples supported by the actual prototype implementation:

- based on a list of resources ontology concepts, select all resources having a "concerns" relation with a given domain ontology concept;
- taking as starting point a domain concept, search all resources concerning that concept;
- search all resources having a relation with a particular resource.

The presence of relations between resources allows the broker agents to dig deeper into a resource knowledge and possibly to infer on it and deduce new knowledge. Knowledge that can improve the description of the resource itself for further search.
Every resource, besides being characterized by its hierarchy and relations with other resources, can be described also by some properties, such as *name*, *description*, *location* and others. For example, a resource such as a web service, can be characterized by the WSDL service descriptor, or the OWL-S file and by a relation like "stub with". The "stub with" relation can in addition specify that the service stub needs the "WSDL2Java", which is a web service resource instance.

Besides query functionality the system allows to add new resources in order to enrich the knowledge base behind Resourceome.

B. The zooming-in zooming-out in the Resourceome

A MAS is generally formed by an agents community operating in one or more distributed platforms. In a distributed environment resources discovery is a very important. To such purpose, maintaining a unique centralize Resourceome acting as resources provider, would be very an impracticable solution as the community growths. On the other hand, a distinct Resourceome for each place, managing only local resources should be reductive, while aligning them would result extremely redundant.

Thanks to the ontologies distribution, we can consider the domain as organized in a top-down model. The resources discovery can be done through several "zoom-in" or "zoom-out" in the different domain ontologies as shown in Figure 3. In the top level of the domain hierarchy can be found one or more upper ontologies concerning very generalized concepts [17]. Zooming-in the domain ontologies, we can navigate the ontologies through the platforms and have a more and more specialized view of our interest domain. These ontologies are in local platforms. The zoom-in is implemented through a *part-of* relations, connecting several domain ontologies. The resources discovery can be executed starting from a general view at a more specific view. In this way it will be possible to discover the needed resources concerning a particular topic of the domain.

V. Case study

The production process of a manufacturing company is usually performed by executing a set of distinct activities, sequential or not. Performing an activity often requires the use of resources. As a case study we have chosen a simply supply chain to build and test electrical domestic appliances (see Figure 5). The supply chain consists of federated enterprises: several suppliers, a production plant, a distribution center and a technical service center. Each enterprise is characterized by a specific role and carries out a set of specific tasks in the virtual organization. Usually, tasks need specific resources like files describing particular tests, components for controlling the machine status during the production process, databases for

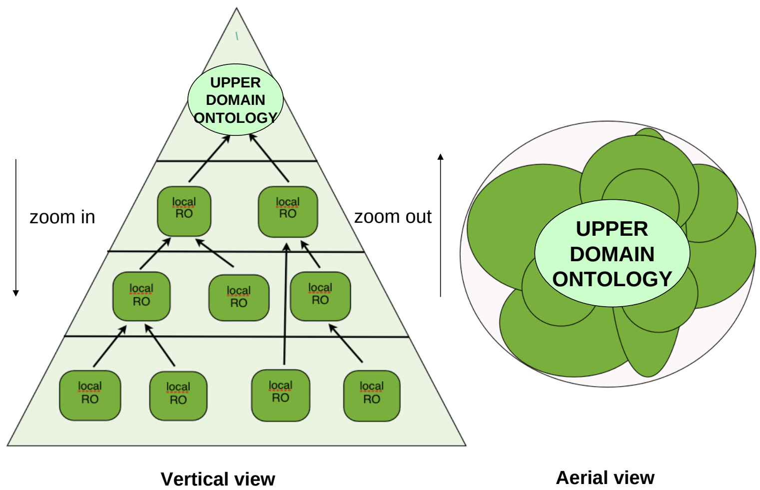

Fig. 3. Domain ontology view

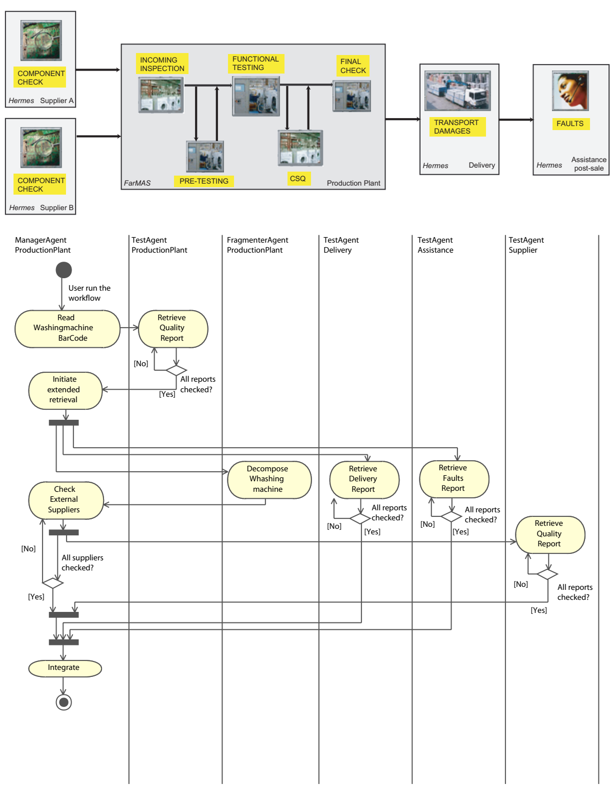

Fig. 4. Case study: the MAS supply chain [18]

managing stores, providers of machines components, software agents to assist the orders management and so forth.

In this context, these and other resources are distributed in the production branches and described by Resourceome (see Figure 2).

In our previous work we have defined a society of autonomous agents created to support the traceability of components information in a federated enterprises environment [18]. A simple supply chain described in Figure 4 for the production of electrical appliances such as washing machines, refrigerators or dishwashers whose componentsÕ traceability is defined in terms of a kind of workflow extended for quality control.

During the testing phase, it could happen that a test shows a malfunction of a particular component of a washing machine, elsewhere assembled. If the same component is not available, it is necessary to find a similar one. Discovering a similar component in the federated enterprises is not an easy task if the semantic description of the component (for us a resource) is not available. In the hypothesis that Resourceome provides the domain ontology for dry washing and the resource ontology for the component of a washing machine, the discovering of a new component could be performed by the cooperation of the enterprise employee, the warehouse agent, the query/answer agents and the store agent.

If the failure occurs frequently, besides blocking the production, this circumstance will cause also inconsistency in the databases distributed in the federation. Also this problem is easily predictable and solvable by a semantic resource management system .

VI. Conclusion

Multi-agent systems can be understood as complex entities where a multitude of agents interact, within a structured environment aiming at some global purpose. The Resourceome system is used to discover the resources necessary for the agents goals achievement. In fact, agents organizations often need heterogeneous resources, such as files, persons, web services or agents providing some services (*service agents*), not of easy to discover. An agent could be blocked because of its

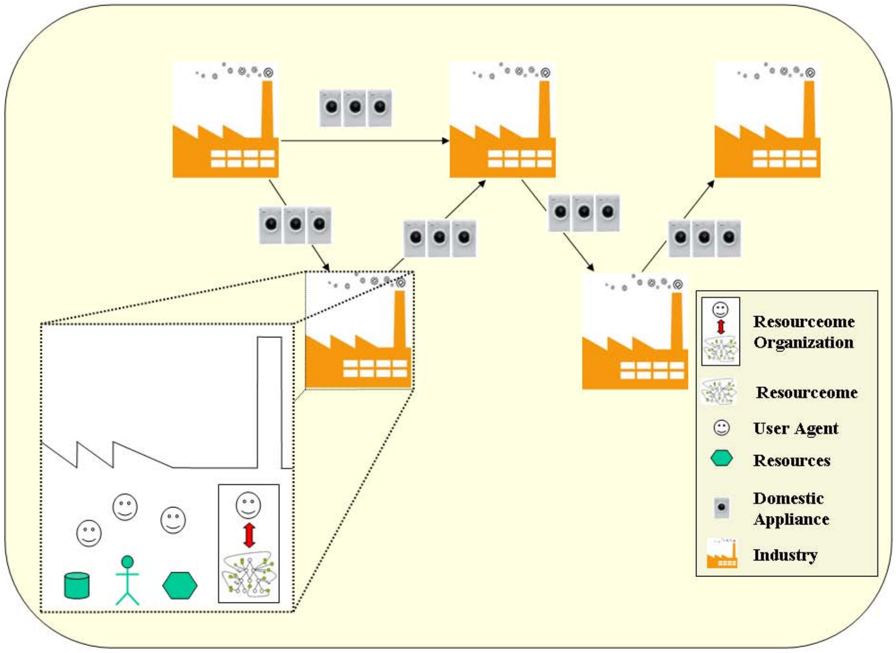

Fig. 5. Case study: the Resourceomic MAS in a supply chain

inability of retrieving a resource or of resource unavailability. The use of a semantic system for resources organization avoids this kind of problem providing analogous resources, replacing the ones searched by the agent, but not available at that moment. So the resources contextualization in a precise domain inside a multi-agent system, represents a very important aspect for the characterization of the environment and its evolutions. In this paper we presented the Resourceome both as semantic system for resources management and contextualization, and as resource broker. The use of ontologies organized as domain, resources and instances ontologies, besides resources discovery, also allows to express roles and conventions of the organization representing the multi-agent system. Since a service agent is itself a resource, the Resourceome can give semantics to Directory Facilitators.

ACKNOWLEDGMENT

The authors would like to thank the Loccioni Group and the *Italian Investment Funds for Basic Research* (MIUR-FIRB) project, *Laboratory of Interdisciplinary Technologies in Bioinformatics* (LITBIO) for supporting this work.

REFERENCES

[1] E. M. N. Cannata and R. B. Altman, "Time to organize the bioinformatics resourceome," *PLoS Comput Biol.*, vol. 1, no. 7:e76, 2005.
[2] M. Klusch and K. Sycara, "Brokering and matchmaking for coordination of agent societies: A survey," in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds. Springer-Verlag, Mar. 2001, ch. 8, pp. 197–224.
[3] N. Cannata, F. Corradini, S. Gabrielli, L. Leoni, E. Merelli, F. Piersigilli, and L. Vito, "Intuitive and machine-understandable representation of the bioinformatics domain and of related resources with resourceomes," in *NETTB: A Semantic Web for Bioinformatics: Goals, Tools, Systems, Applications*, June 2007.
[4] T. B.-L. et al., "Rfc 2396 - uniform resource identifiers (uri): Generic syntax," 1998, `www.faqs.org/rfcs/rfc2396.html`.
[5] T. Clark, S. Martin, and T. Liefeld, "Globally distributed object identification for biological knowledgebases." *Briefings in Bioinformatics*, vol. 5, no. 1, pp. 59–70, 2004.
[6] "The digital object identifier system," `http://doi.org`.
[7] L. O. and S. R. R., "Resource description framework (rdf): Model and syntax specification," 1999, `www.w3.org/TR/REC-rdf-syntax`.
[8] `www.fipa.org/specs/fipa00001/SC00001L.pdf`.
[9] "OWL web ontology language guide," `www.w3.org/TR/owl-guide/`.
[10] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
[11] N. R. Jennings, "An agent-based approach for building complex software systems." *Commun. ACM*, vol. 44, no. 4, pp. 35–41, 2001.
[12] "Sparql query language for rdf," `www.w3.org/TR/rdf-sparql-query/`, 2007.
[13] J. D. Wren, "404 not found: the stability and persistence of urls published in medline." *Bioinformatics*, vol. 20, no. 5, pp. 668–672, 2004.
[14] M. V. M. Colombetti, N. Fornara, "Linguaggio e realtà sociale nei sistemi di agenti artificiali," *Networks*, no. 1, pp. 48–67, 2003.
[15] K. P. Sycara, J. A. Giampapa, B. K. Langley, and M. Paolucci, "The RETSINA MAS, a case study." in *SELMAS*, 2002, pp. 232–250.
[16] J. R. Searle, *The construction of social reality*, simon & schuster ed., 1995.
[17] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with DOLCE." in *EKAW*, 2002, pp. 166–181.
[18] D. Bonura, F. Corradini, E. Merelli, and G. Romiti, "Farmas: a MAS for extended quality workflow," in *2nd IEEE International Workshop on Theory and Practice of Open Computational Systems*. IEEE Computer Society Press, 2004.