Agent-Grid Integration Ontology

Frédéric Duvert, Clement Jonquet, Pascal Dugénie, Stefano A. Cerri

Agent-Grid Integration Ontology

Frederic Duvert, Clement Jonquet, Pascal Dugenie, and Stefano A. Cerri

LIRMM, CNRS & University Montpellier 2
161 Rue Ada, 34392 Montpellier Cedex 5, France
{duvert, jonquet, dugenie, cerri}@lirmm.fr

Abstract. The integration of GRID and MAS (Multi-Agents Systems) is an active research topic. We have recently proposed the Agent-Grid Integration Language, to describe a service-based integration of GRID and MAS models. However, the complexity of the mutual integration aspects leads us to define a rigorous way to formalize the key concepts, their relations and the integration rules by means of an ontology. With this ontology, we can describe the elements and their composition that occur in various service exchange scenarios with agent on the Grid. The ontology could be used both to model the behaviour of GRID-MAS integrated systems and to check the consistency of these systems and their instances. A concrete scenario is illustrated.

1 Introduction

GRID and MAS (Multi-Agent Systems) are two kinds of distributed systems. Yet, the motivations are different. GRID focuses on a reliable and secure resource-sharing infrastructure, whereas MAS focuse on flexible and autonomous collaboration and problem solving in uncertain and dynamic open environments. The GRID and MAS communities believe in the potential of GRID and MAS to enhance each other because these models have developed significant complementarities [1]. In [2,3] we explain why these two domains join with the concept of service:

– GRID and MAS have each developed a service oriented behaviour, therefore the concept of service may represent a common integration;
– New needs in service exchange scenarios are clearly highlighted (dynamicity, composition, conversation based, user-centred behaviour, business processes, semantics, etc.) [4] and may be met by integrating GRID and MAS complementarities.

One of the crucial explorations concerns the substitution by an agent-oriented kernel of the current object-oriented kernel of services available in GRID. In [2,3] we propose a model for GRID-MAS integrated systems which considers that services are exchanged (i.e., provided and used) by *agents* through *GRID* mechanisms and infrastructure. In this model, concepts, relations between them and rules of these systems are semantically described by a set-theory formalization and a common graphical description language, called *Agent-Grid Integration Language* (AGIL).

In this paper, we formalize, by means of an ontology the GRID-MAS integration model proposed by AGIL. This ontology, called *Agent-Grid Integration Ontology* (AGIO), describes the semantics of GRID-MAS integrated system elements as well

as the behaviour of GRID-MAS integrated systems. AGIO's GRID concepts are directly influenced by OGSA (Open Grid Service Architecture) [5] and AGIO's MAS concepts are influenced by different approaches in MAS, such as the STROBE model [6], the Agent-Group-Role (AGR) model [7], Belief-Desire-Intention (BDI) architectures, Foundation for Intelligent Physical Agents (FIPA) agents, or other elements of the MAS literature [8,9]. AGIO describes the elements implied in service-oriented interaction between agents. It is actually a meta-description, allowing agents to agree on what they are, what is a service, a host, etc.

2   GRID-MAS Integrated Model

Service-based integration of GRID and MAS models. The concept of service is clearly at the intersection of the GRID and MAS domains. GRID is said to be the first distributed architecture (and infrastructure) really developed in a service-oriented perspective: Grid services are compliant Web services, based on the dynamic allocation of virtualized resources to an instantiated service [5]. Whereas Web services have instances that are stateless and persistent, Grid service instances can be either stateful or stateless, and can be either transient or persistent. A stateful service has an internal state that persists over multiple interactions. For a recent precise overview of Grid service concepts and standardization, see for example [10].

On the other hand, agents are said to be autonomous, intelligent and interactive entities who may use and provide services (in the sense of particular problem-solving capabilities) [8,9]. Actually they have many interesting characteristics for service exchange: they are reactive, efficient, adaptive, they know about themselves, they have a memory and a persistent state, they are able to have conversation, work collaboratively, negotiate, learn and reason to evolve, deal with semantics associated to concepts by processing ontologies, etc. MAS and service-oriented computing recently turned to one another considering the important abilities of agents for providing and using dynamic composed/composite services, semantic services, business processes, etc. [4].

Key GRID and MAS concepts and their integration. GRID is a resource-sharing system. Grid resources are contributed by *hosts*. A host is either a direct association between a *computing resource* and a *storage resource* or a *host coupling*. The sharing of these resources is implemented by the virtualization and the reification of these resources in *service containers*. A Grid *service* is included in a hosting environment in order to exist and to evolve with their own private contexts (i.e., set of resources). This is the role of the service container which is the reification of a portion of the virtualized resource available in a secure and reliable manner. A service container contains several types of services. A service may instantiate another service in the same or different service container. Each service is identified by a *handle*. Since a container is a particular kind of service, it is created either through the use of a service factory or by the direct core GRID functionality. A service container is allocated to (and created for) one and only one group of *agents*,[1] called a *Virtual Organization* (VO).[2] Each agent may belong to

---

[1] The term agent is used to uniformly denote artificial agent, human agent and Grid user.

[2] The term VO unifies the concept of VO in GRID and the concept of group in MAS.

several VOs. The relation between VO members and Grid services in the associated container is embodied by a *Community Authorization Service* (CAS) which formalizes the VO-dedicated policies of service by members. In order to participate in GRID, hosts and agents must hold a *X509 certificate* signed by a special authority.

An *agent* possesses both intelligent and functional abilities. These are represented respectively by the agent *brain* and *body*. The brain is composed of a set of rules and algorithms (e.g., machine learning) that give to the agent learning and reasoning skills. It also contains the agent knowledge, objectives, and mental states (e.g., BDI). The body is composed of a set of *capabilities* which correspond to the agent's capacity or ability to do something, i.e., to perform some task. These capabilities may be interfaced as Grid services in the service container that belongs to a VO an agent is a member of. In the agent's body, these capabilities may be executed in a particular context called a *cognitive environment*.[3] A cognitive environment contains several capacities. An agent may have several cognitive environments which correspond to the different conversation contexts and languages it develops by interaction with other agents. These interactions can be for example service exchanges i.e., situations where agents use the service another agent provides.

The GRID-MAS integrated model is illustrated in Fig. 1. We sum-up here the two main underlying ideas:

- The representation of agent capabilities as Grid services in a service container, i.e., viewing Grid service as an 'allocated interface' of an agent capability by substituting the object-oriented kernel of Web/Grid services with and agent oriented one;
- The assimilation of the service instantiation mechanism – fundamental in GRID as it allows Grid services to be stateful and dynamic – with the dedicated cognitive environment instantiation mechanism – fundamental in STROBE as it allows one agent to dedicate to another one a conversation context.

3   GRID-MAS Integration Ontology

3.1   Ontology Modelling

When formalized and computerized, shared knowledge can serve as the basis for better understanding among agents. In recent years, the representation of such shared knowledge has largely been implemented by ontologies i.e., formal, computerized conceptualization of the notions, properties and relationships in a domain [11]. Nowadays, ontologies are used by agents each time a semantic description is needed. Ontologies are composed of *concepts*, *relations* and *instances*. For example, if you want to define a car, you should say: 'a car is a transportation object, with four wheels, and you need a licence to drive it. MyCar is a car.' 'Car' is a concept, 'is a' is a relation, and 'MyCar' is an

---

[3] Conversations and their states are represented in the STROBE model [6] by cognitive environments. We do not detail this aspect here. In other agent architectures, cognitive environments may simply be viewed as conversation contexts. Our GRID-MAS integrated model, was influenced by the STROBE model as it is a communication and representation agent model developed in a service perspective that fits well Grid service mechanisms such as for example the concept of instantiation.
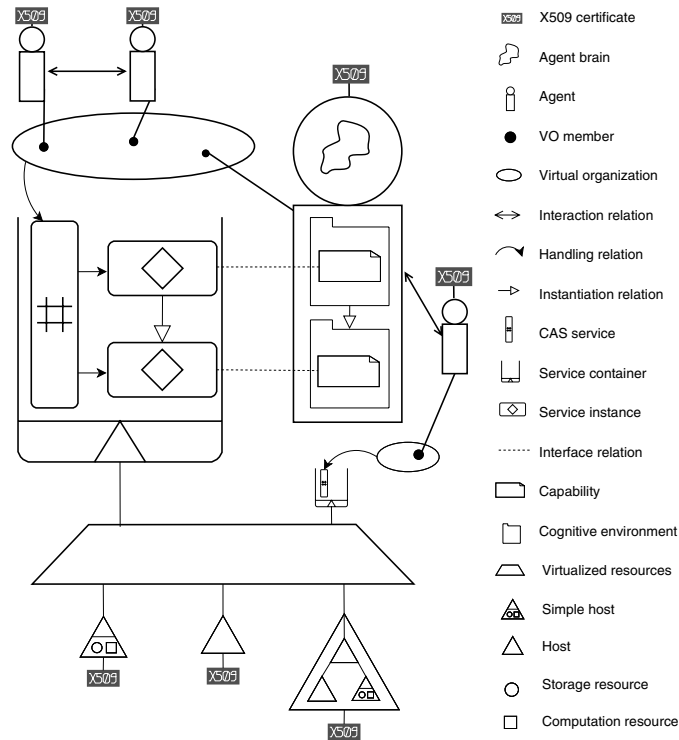
The legend items for the figure are:

- X509 certificate
- Agent brain
- Agent
- VO member
- Virtual organization
- Interaction relation
- Handling relation
- Instantiation relation
- CAS service
- Service container
- Service instance
- Interface relation
- Capability
- Cognitive environment
- Virtualized resources
- Simple host
- Host
- Storage resource
- Computation resource

Fig. 1. The GRID-MAS integrated model described in AGIL's graphical description language

instance. Ontologies allow also to define *rules*. A rule is a constraint on a set of relations, that is not directly expressed by the relations. In the rest of the paper, we use the following writing conventions: `Concept`, *Relation* and INSTANCE e.g., MYCAR *is a* Car.

A common standard for representing ontology is the Web Ontology Language (OWL) [12]. Ontologies editors free the designer from writing directly XML/OWL code. They automatically generate it, allowing ontologies to be both understandable and processable by human and artificial agents. The Protégé knowledge-modelling environment (*http://protege.stanford.edu*) is a de facto standard tool that supports single and multi-user construction of ontologies. It enable designers to express rules by means of the Semantic Web Rule Language (SWRL) [13].

3.2 Agent-Grid Integration Ontology

GRID and MAS key concepts simplified and briefly summarized in Sect. 2 are translated into an ontology. Concepts are presented in Table 1 and relations in Table 2. The integration rules are expressed by means of SWRL rules. AGIO is a procedural ontology that describes the different elements of GRID-MAS integrated systems and their interactions. Adding an instance of a GRID-MAS integrated system to AGIO would make it evolve as a descriptive ontology.

Table 1. AGIO's concepts and inheritence

CONCEPT	INHERITS
Resource	Thing
ComputingResource	Resource
StorageResource	Resource
Host	Thing
HostSimple	Host
HostCoupling	Host
VirtualizedResource	Thing
X509	Thing
Service	Thing

CONCEPT	INHERITS
ServiceContainer	Service
CAS	Service
NormalService	Service
Agent	Thing
HumanAgent	Agent
ArtificialAgent	Agent
VO	Thing
CognitiveEnv	Thing
Capacity	Thing

Using an ontology to describe GRID-MAS integrated models is interesting because we can describe using the same formalization both the model and its instances. Designers of GRID-MAS integrated models may instantiate AGIO concepts in order to formalize their systems and check their consistency thanks to AGIO rules, as it is illustrated in Sect. 4.

One pre-requisite to any collaboration is to have mutual understanding about things the collaboration is dealing with. The intrinsic elements agents need a mutual understanding about, are themselves and the world in which they evolve. It is called a meta-description. Actually, AGIO is a language for expressing such a meta-description because agents of GRID-MAS integrated systems can use it in order to represent the world in which they exist and exchange services. AGIO allows agents to agree on what they are, what is a service, a host, etc.

Table 2. AGIO' relations and types

RELATION	TYPE	DOMAIN	RANGE
couples	relation	HostCoupling	Host
virtualizes	relation	VirtualizedResource	Host
reifies	relation	ServiceContainer	VirtualizedResource
holds	relation	Agent ∪ Host	X509
belongs	relation	Agent	VO
interacts	function	Agent	Agent
uses	relation	Agent	Service
provides	function	Agent	Service
exchanges	relation	Agent	Agent
instantiates	relation	NormalService CognitiveEnv	Service CognitiveEnv
includes	function	ServiceContainer	CAS ∪ NormalService
handles	relation	VO CAS	CAS CAS ∪ NormalService
interfaces	function	Service	Capability
executes	function	CognitiveEnv	Capability
owns	function	ArtificialAgent	CognitiveEnv

3.3 OWL/Protégé Implementation

AGIO was implemented in Protégé. For instance, the class[4] `Agent` and the property *interfaces* are defined in OWL as:

```
<owl:Class rdf:about="#Agent">
    <owl:disjointWith rdf:resource="#X509"/>
    ...
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="belongs"/>
        </owl:onProperty>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#VO"/>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    ...
    <owl:ObjectProperty rdf:about="#interfaces">
        <owl:inverseOf rdf:resource="#isInterfacedBy"/>
        <rdfs:range rdf:resource="#Capability"/>
        <rdfs:domain rdf:resource="#Service"/>
    </owl:ObjectProperty>
```

An example of integration rule may be: *an agent must hold a X509 certificate in order to be member of a VO*. It is expressed in SWRL as:

$$Agent(?x) \land X509(?y) \land VO(?z) \land \textbf{belongs (?x,?z)} \rightarrow holds(?x, ?y)$$

4 Example of Scenario with AGIO

Data mining scenario. [14] proposes a Grid based architecture for supporting knowledge discovery processes such as data mining, data analysis, etc. The *Knowledge Grid* is a set of layers upon the GRID, allowing parallel and distributed knowledge discovery. This knowledge Grid architecture is based on a set of services constructed on the top of basic core GRID services. In this section, we present a "data mining" scenario which considers an artificial agent that provides users with a data mining Knowledge Grid service. This is a good example of a potential GRID-MAS integrated system, because Knowledge Grid services are typically the kind of services that could be very enhanced by intelligent agent abilities.

AGIO's instances of the scenario. Let us consider the following elements: BOB, LUCY and DMAGENT are three instances of `Agent`. They *belong* VO1, an instance of VO (Virtual Organization). They *hold* respectively XB, XL, and XD, three instances of X509. DMAGENT *provides* DMSFACT, DMSB and DMSL, three instances of

---

[4] Protégé calls class and property respectively concept and relation.

`NormalService`. They are *interfaced* by respectively DMGENERICCAP, DMcB and DMcL, and they are *executed* in CEGLOBAL, CELOCALB and CELOCALL (Cognitive Environment), all three *owned* by DMAGENT. DMsFACT *instantiates* DMsB and DMsL and CEGLOBAL *instantiates* CELOCALB and CELOCALL. CAS1, an instance of `CAS`, *handles* itself, DMsFACT, DMsB, DMsL and is *handled* by VO1. All these services are *included* in SC1, an instance of `ServiceContainer`. The rest of the instances (`VirtualizedResource`, `Host`, etc.) are not specified here. In particular, the `VirtualizedResource` available for DMsFACT, DMsB and DMsL may come from a set of different `StorageResource` that DMAGENT integrates and process via its data-mining services.

BOB wants to analyze some elements in the common shared set of data proposed by GRID. LUCY looks for a specific data pattern in same set of data. They both have different objectives. Thus, BOB and LUCY respectively *uses* DMsB and DMsL. These instances are customized, thank to DMAGENT intelligent and interactive abilities. The two different interfaced capabilities are executed in two different conversation contexts dedicated for their users. These Knowledge Grid services are significantly enhanced by agent abilities. [6,3] detail the ways an agent can dynamically generate services to another one, following a conversation.

Consistency. On a given scenario such as the "data-mining" scenario, one can check the consistency of the set of instances thanks to the SWRL rules implemented in AGIO e.g., if exists the right set of instances for a given property. Each conjunction may be considered as a predicate. For example:

$$Agent(\text{DMAGENT}) \wedge CAS(\text{CAS1}) \wedge Service(\text{DMsB}) \wedge$$
$$VO(\text{VO1}) \wedge CognitiveEnv(CELocalB) \wedge Capability(DMcB) \wedge$$
$$\textbf{provides(DMAGENT,DMsB)} \rightarrow$$
$$owns(\text{DMAGENT}, CELocalB) \wedge executes(CELocalB, DMcB) \wedge$$
$$interfaces(\text{DMsB}, DMcB) \wedge belongs(\text{DMAGENT}, \text{VO1}) \wedge$$
$$handles(\text{VO1}, \text{CAS1}) \wedge handles(\text{CAS1}, \text{DMsB})$$

5   Related Work and Discussion

There is an increasing amount of research activity in GRID and MAS convergence taking place.[5] The Control of Agent-Based Systems (CoABS) project [15], proposed in 1999 by the DARPA, is the first research initiative in GRID-MAS integration. In this project, priority was given to GRID development, but the participants already envisage a combination of GRID and MAS domains. The use of agents for GRID was very early suggested in [16]. The authors specifically detail how agents can provide a useful abstraction at the Computational Grid layer. MAS has also been established as a key element of the Semantic Grid [17]. More recently, the why GRID and MAS need each other as been established by [1].

---

[5] See, for example, *Agent-Based Cluster and Grid Computing* workshops, *Smart Grid Technologies* workshops, the *Multi-Agent and Grid Systems* international journal.

Using MAS principles to improve core GRID functionalities represent the main part of related work. For example, MAS-based GRID approaches for resource management use agents for an effective management of the vast amount of resources that are made available within a GRID environment as they have, for example, excellent trading and negotiation abilities (negotiation between resource agents and allocator agent) e.g., [15,18,19]. Moreover MAS can use the Grid Laboratory Uniform Environment (GLUE Schema[6]) to access easily to resources. The GLUE Schema is a description and a representation of GRID heterogeneous resources. Agents, using this GLUE Schema, are able to retrieve resources without taking into account the resources origin. It would be a good way to share the status and availability of resources and resources themselves.

Another example is the use of MAS for VO management i.e., formation, operation and dissolution of VOs. The main work is this domain is the Grid-enabled Constraint-Oriented Negotiation in an Open Information Services Environment (CONOISE-G) project [20].

Some work has also been proposed in using agents to enhance Web/Grid services or integrating the two approaches. In particular, in order to connect MAS communication approaches with business process management approaches e.g., [21,22,23,4]

However, none of these works propose a real integration of MAS and GRID. Rather, they focus on how MAS and AI techniques may enhance core GRID functionalities. Our vision of a GRID-MAS integration is not a simple interoperation of the technologies (it goes beyond a simple use of one technology to enhance the other). Besides, describing the integration of GRID and MAS by means of an ontology sets a new formal foundations to the integrating ideas.

AGIO ontology focuses on modelling an integration of GRID-MAS. We can use the OntoGrid[7] to formalize more precisely GRID concept, but the aim of our work is to present an integration of GRID-MAS using general concepts of GRID. We do not have the ambition to represent all concepts linked to GRID, but to succintly express the way that MAS and GRID are integrated each other.

The advantages of the GRID-MAS integrated model formalized by AGIO are precisely detailed in [3]. We briefly describe some of them here. There is no real standard in the MAS community to describe agent capabilities. Interfacing them as Grid services is thus a potential step towards standardization. This integrated model does not restrict MAS or GRID in any way. Everything feasible with MAS or GRID today still holds. VO management benefits from both GRID and MAS organizational structure formalism, e.g., Agent-Group-Role [7], CAS service, X509 certificate, etc. Service exchange benefits from the important agent communication abilities, e.g., dealing with semantics, ability to have a conversation, etc. The challenge of modelling dynamic agent conversations becomes the same as the owe of dynamically composing and choreographing services in business processes. The model subsumes a significant number of the MAS-based GRID approaches mentioned before thanks to the reflexivity of GRID, which defines some GRID core functionalities as (meta-)Grid services (e.g., service container, CAS). Therefore, GRID and MAS would appreciate a common ontology which:

---

[6] *http://glueschema.forge.cnaf.infn.it/*

[7] *http://www.ontogrid.net*

– describes simply and clearly key concepts and their integration;[8]
– uses the same terms and representations for an identical concept e.g., VO and group, choreography of service and agent conversation, role and service.
– rigorously fixes the integration rule;
– may help researchers of GRID and MAS communities to specify and model their GRID-MAS integrated applications and systems by instantiating AGIO concepts (pass from a procedural ontology to a descriptive one);
– would promulgate the development of GRID-MAS integrated systems by proposing a uniform way of describing GRID and MAS together.

In a sake of simplicity, the paper presents only some part AGIO's rules or OWL elements. We invite the reader to refer to [24], for a complete specification of AGIO, or to [3] for a complete specification of AGIL.

6   Conclusion and Perspectives

Even if using agents for GRID was very early suggested [15,16,17], Foster et al. [1] propose the real first step in GRID-MAS integration as it examines work in these two domains, firstly to communicate to each community what has been done by the other, and secondly to identify opportunities for cross fertilization as they explained how GRID and MAS developed significant complementarities. The work proposed by [2], [3] and this paper suggests a second step by proposing a GRID-MAS integrated model. In particular, in this paper, we formalize this model by means of an ontology. AGIO describes concepts of the GRID-MAS integrated model, as well as relations between each of these concepts. Moreover, relations are strengthened by SWRL rules which guarantee coherence.

Some perspectives for the future of AGIO may be:

– To add new concepts, properties and rules according of the evolution of the GRID-MAS integrated model e.g., different types of X509 certificates (proxy's, certification authority's);
– To connect AGIO with other meta-ontologies used to define service, GRID or MAS concepts;
– To experiment the ontology with large-scale GRID-MAS integrated systems;
– To integrate in AGIO other different agent representation approaches;
– To classify services with their intrinsic properties (e.g., statefulness, transient, multipoint);
– To define tools that assist designers by automatically checking, during system development, the consistency of the ontology.

References

1. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: why Grid and agents need each other. In: 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'04. Volume 1., New York, NY, USA (2004) 8–15

---

[8] In particular, GRID needs a description language that summarizes and rigorously explains GRID concepts. Without considering the agent side, AGIO may play also this role.

2. Jonquet, C., Dugenie, P., Cerri, S.A.: Service-based integration of Grid and multi-agent systems models. Research report 06012, University Montpellier II, France (2006) www.lirmm.fr/˜ jonquet/Publications.

3. Jonquet, C., Dugenie, P., Cerri, S.A.: AGIL specifications. Research report 06030, University Montpellier II, France (2006) www.lirmm.fr/˜ jonquet/Publications.

4. Singh, M.P., Huhns, M.N.: Service-Oriented Computing, Semantics, processes, agents. John Wiley & Sons (2005)

5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the Grid: an Open Grid Services Architecture for distributed systems integration. In: Open Grid Service Infrastructure WG, Global Grid Forum, The Globus Alliance (2002)

6. Jonquet, C., Cerri, S.A.: The STROBE model: Dynamic Service Generation on the Grid. Applied Artificial Intelligence, Special issue on Learning Grid Services **19**(9-10) (2005) 967–1013

7. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: an organizational view of multi-agent systems. In Giorgini, P., Müller, J.P., Odell, J., eds.: 4th International Workshop on Agent-Oriented Software Engineering, AOSE'03. Volume 2935 of Lecture Notes in Computer Science. Springer-Verlag, Melbourne, Australia (2003) 214–230

8. Ferber, J.: Multi-agent systems: an introduction to distributed artificial intelligence. Addison Wesley Longman, Harlow, UK (1999)

9. Wooldridge, M.: An introduction to multiagent systems. John Wiley & Sons, Chichester, UK (2002)

10. Comito, C., Talia, D., Trunfio, P.: Grid services: principles, implementations and use. Web and Grid Services **1**(1) (2005) 48–68

11. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5**(2) (1993) 199–220

12. : OWL Web Ontology Language use cases and requirements. W3c recommendation, World Wide Web Consortium (2004)

13. O'Connor, M., Knublauch, H., Tu, S.W., Grosof, B.N., Dean, M., Grosso, W.E., Musen, M.A.: Supporting Rule System Interoperability on the Semantic Web with SWRL. In Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A., eds.: 4th International Semantic Web Conference, ISWC'05. Volume 3729 of Lecture Note in Computer Science., Galway, Ireland, Springer-Verlag (2005) 974–986

14. Cannataro, M., Talia, D.: The Knowledge Grid. Communications of the ACM **46**(1) (2003) 89–93

15. Manola, F., Thompson, C.: Characterizing the agent Grid. Technical report 990623, Object Services and Consulting, Inc. (1999)

16. Rana, O.F., Moreau, L.: Issues in building agent based computational Grids. In: 3rd Workshop of the UK Special Interest Group on Multi-Agent Systems, UKMAS'00, Oxford, UK (2000)

17. Roure, D.D., Jennings, N., Shadbolt, N.: Research agenda for the Semantic Grid: a future e-science infrastructure. Technical report, University of Southampton, UK (2001) Report commissioned for EPSRC/DTI Core e-Science Programme.

18. Martino, B.D., Rana, O.F.: Grid performance and resource management using mobile agents. In Getov, V., Gerndt, M., Hoisie, A., Malony, A., Miller, B., eds.: Performance Analysis and Grid Computing. Kluwer (2003) 251–263

19. Cao, J., Spooner, D.P., Jarvis, S.A., Nudd, G.R.: Grid load balancing using intelligent agents. Future Generation Computer Systems **21**(1) (2005) 135–149

20. Patel, J., Teacy, W.T.L., Jennings, N.R., Luck, M., Chalmers, S., Oren, N., Norman, T.J., Preece, A., Gray, P.M.D., Shercliff, G., Stockreisser, P.J., Shao, J., Gray, W.A., Fiddian, N.J., Thompson, S.: Agent-based virtual organisations for the Grid. Multiagent and Grid Systems **1**(4) (2005) 237–249

21. Buhler, P.A., Vidal, J.M., Verhagen, H.: Adaptive workflow = Web services + agents. In: International Conference on Web Services, ICWS'03, Las Vegas, NV, USA, CSREA Press (2003) 131–137
22. Ardissono, L., Goy, A., Petrone, G.: Enabling conversations with Web services. In: 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'03, Melbourne, Australia, ACM Press (2003) 819–826
23. Greenwood, D., Calisti, M.: Engineering Web service - agent integration. In: IEEE Systems, Cybernetics and Man Conference, The Hague, Netherlands, IEEE Computer Society (2004)
24. Duvert, F.: An ontology of Grid and Multi-agent systems integration. Master's thesis, University Montpellier 2, Montpellier, France (2006)