A Semantic Collaborative Awareness Model to deal with Resource Sharing in Grids

Manuel Salvadores[1,2],Pilar Herrero[2], José Luis Bosque[3], María S. Peréz[2]

[1]*IAM, School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK*
ms8@ecs.soton.ac.uk

[2]*Universidad Politécnica de Madrid, Facultad de Informática, Campus de Montegancedo, Boadilla del Monte, Madrid, Spain*
pherrero@fi.upm.es, mperez@fi.upm.es

[3]*Departamento de Electrónica y Computadores Universidad de Cantabria*
joseluis.bosque@unican.es

Abstract

Grid Computing environments are mainly created to lead the shared use of different resources based on business/science needs. The way these resources are shared in terms of CPU cycles, storage capacity, software licenses, ... is normally established by the availability of these resources out of the local administration context. Semantic Grid is the extension of Grid Computing with Semantic Web based technologies. Semantic Web allows grid management data to be machine-understable represented, therefore reasoning can handled complicated situations in Virtual Organization management. This paper presents the extension of CAM (Collaborative Awareness Model) to manage Virtual Organizations in Semantic Grid environments. CAM applies some theoretical principles of awareness models to promote resource interaction and management as well as task delivery.

Key words: Grid Computing, Ontology, Collaborative, Awareness, Semantic Grid

1 Introduction

Some challenging questions come up when resource sharing wants to be achieved over complex structures of organizations. Questions like *What are the terms over which I would like to leave others to use my resources?* or *What are the conditions for using third parties computational power?* are well known issues to be faced in Grid Computing environments. Grid Computing gathers the solutions to these kind of questions around the concept of Virtual Organization (VO) [11]. A VO is a widely spread concept used in several fields of Computer Science (e.g. Agents Collaboration and Negotiation protocols, Collaborative Tools and Social Networks). Currently, from the point of view of Grid Computing, a VO is commonly defined as *"a group of individuals or institutions who share the computing resources of a grid for a common goal"* [10,9]. Efforts done by the Open Grid Forum[2] has revealed that managing a grid system thus its VOs is highly complex [3].

The way CAM deals with VO management is based on previous Computer Supported Cooperative Work (CSCW) research [5]. CSCW are characterized by their ability to support and manage large numbers of coordinated heterogeneous resources and services while they cooperate to accomplish a common goal [4]. The CSCW paradigm has traditionally encompassed distributed systems technologies such as middleware, business process management and web technologies. CAM has been designed, from the beginning to be a parametrical, generic, open, model that could be extended and easily adapted. This model allows managing not just resources and information but also interaction and awareness. More specifically, CAM allows: i) controlling the user interaction; ii) guiding the awareness towards specific users and resources; iii) scaling interaction through the awareness concept.

The Semantic Web is an extension of the current Web where data can be processed by humans as well as machines to find, share and integrate information more easily [6]. To date, the Web is mainly designed for human use and meaning can be hardly interpreted by software agents. With Semantic Web technologies, resources (like web pages) can be augmented with semantic metadata which can be read and used by software agents.

After next section which exposes the research associated to this area work the paper introduces CAM's key concepts to later develop the backbone of this research: *CAM's Semantic Web Implementation*, to end up with the architecture and conclusions.

2 Related Work

To date, as far as we know, there are not any awareness-based systems used to deal with the management of resources in Grid Computing environment. However, there are solutions dealing with VO management in the Semantic Grid research field. An extension to the Open Grid Services Architecture (OGSA) [9] for the Semantic Grid is presented in [8] as S-OGSA. S-OGSA came up as an architecture for the Semantic Grid providing high level capabilities and the functionalities that should be implemented for deploying a “standard” Semantic Grid application.

S-OGSA introduces the awareness concept by the definition of “Semantically Aware Grid Services”, this term is introduced to named services that potentially can consume Semantic Bindings. Semantic Bindings are the different management actions, not the service execution itself, that can be performed based on the metadata provided during the service-client interaction. These actions could involve functionalities such as VO authorization, search over semantic service catalogue or ontology service modifications. It is important to state that the awareness presented in S-OGSA through the “Semantically Aware Grid Services” is high level definition and it does not provide low definitions for the awareness model, as CAM does.

Previous research on CAM has set up the key concepts to manage awareness of interactions [13], task delivery, load balancing and dispatching [14] and self-configuration over rule management [16] as well as model validation on different scenarios [15,14]. As it will be described along the rest of paper this research adds SW capabilities to the previous research done in CAM.

3 Spatial Model of Interaction and Collaborative Awareness Model

CAM, which allows to manage awareness in collaborative grid environments, has been designed based on the extension and reinterpretation of the Spatial Model of Interaction (SMI) [5], an awareness model designed for CSCW. This reinterpretation, open and flexible enough, merges the basic ideas behind the Open Grid Services Architecture (OGSA) [9] features with theoretical principles and theories of multi-agents systems, to create a collaborative and cooperative grid environment within which it is possible to manage different levels of awareness.

CAM relies on a set of key concepts, some of them coming from SMI, and it allows the interaction in environments where a spatial metric can be identified [5]. Given a grid environment containing a set of resources and a T task which needs to be solved in this environment, if this task is made up by a set of processes $T = \sum_{i=0}^{n} p_i$. Where p_i are the processes needed to solve task T in the system. Moreover, these processes could be related to power, disk space, data and/or applications. CAM intends to solve task T in a collaborative and, if possible, cooperative way, by means of a set of key concepts:

- **Focus** is the subset of the space on which the user has focused his attention with the aim of interacting with. This selection will be based on different parameters and characteristics, such as power, disk space, data and/or applications, depending on the T requirements.

- **Nimbus** is the tuple $(NimbusState, NimbusSpace)$ containing information about:
  · the state of the system in a given time (NimbusState);
  · the subset of the space in which a given resource projects its presence (NimbusSpace).

  As for the state of system (NimbusState), the "projection" of this state will present different resource's properties, such as load of the system, disk space, data information stored/processed, processes/applications to carry out, etc. For each of these characteristics the NimbusState could have three possible values: Null, Medium or Maximum. The NimbusState gets the Maximum value when the node has at its disposal all its resources to solve the T task, Medium if the node has at its disposal only a part of its resources to solve the T task, and Null if the node has not resources at its disposal to solve the T task. The NimbusSpace will determine those machines that could be taking into account in the collaborative process.
- **Unidirectional Awareness of Interaction** $AwareInt_{R_1 \Rightarrow R_2}$ This concept will quantify the degree, nature or quality of asynchronous unidirectional interaction between distributed resources. Following the awareness classification introduced by Greenhalgh in [12], this awareness could be Full, Peripheral or Null, according to the these rules:

  $AwareInt_{A \Rightarrow B} = Full$ if and only if $B \in Focus(A) \wedge A \in Nimbus(B)$

  $$AwareInt_{A \Rightarrow B} = Peripheral \text{ if and only if } \begin{cases} B \in Focus(A) \wedge A \notin Nimbus(B) \\ \vee \\ B \notin Focus(A) \wedge A \in Nimbus(B) \end{cases}$$

  $AwareInt_{A \Rightarrow B} = Null$ if and only if $AwareInt_{A \Rightarrow B} \neq Full \wedge AwareInt_{A \Rightarrow B} \neq Peripheral$
- **Bidirectional Awareness of Interaction (AwareInt)** This concept will quantify the degree, nature or quality of asynchronous bidirectional interaction between distributed resources.

  $(AwareInt_{A \Rightarrow B} = Full \wedge AwareInt_{B \Rightarrow A} = Full) \Rightarrow AwareInt(A, B) = true$

  $(AwareInt_{A \Rightarrow B} \neq Full \vee AwareInt_{B \Rightarrow A} \neq Full) \Rightarrow AwareInt(A, B) = false$
- **Aura** Sub-space which effectively bounds the presence of a resource within a given medium and which acts as an enabler of potential interaction. It can delimit and/or modify the focus, the nimbus (NimbusSpace) and therefore the awareness.

CAM, moreover, extends the key concepts of the SMI, introduce new concepts such us:

- **Interactive Pool** This function returns the set of resources interacting with a given resource.
  $AwareInt_{A \Rightarrow B} = Full \Rightarrow B \in InteractivePool(A)$
- **Task Resolution** This function determines if there is a service in the resource B, being $NimbusState(B) \neq Null$, such that could be useful to execute T (or at least one of its processes P_i).
  $TaskResolution(B, T) = \{(p_i, s)\}$
  Where s is the score to carry out pi in the B resource, being its value within the range $[0, \infty)$: 0 if the B resource fulfils all the minimum requirements to carry out the process pi; once the B resource fulfils the all the minimum requirements to carry out the process pi, the higher is the surplus over these requirements, the higher will be the value of this score.
- **Virtual Organization** This function will take into account the set of resources determined by the Interactive Pool function and will return only those in which it is more suitable to execute the task T (or at least one of its processes p_i). This selection will be made by means of the TaskResolution function.
  $AwareInt_{A \Rightarrow B} = Full \Rightarrow B \in InteractivePool(A)\, TaskResolution(B, T) = \{(p_i, s)\} \Rightarrow B \in VirtualOrganization(A, T)$

Resources belonging to this VO could access the resources, as they are aware of them, to solve specific problems, and they could collaborate each other, getting therefore a VO. Collaboration is broadly defined as the interaction among two or more individuals and can encompass a variety of behaviours, including communication, information sharing, coordination, cooperation, problem solving, and negotiation.

As far as we know, none of the last Web Services (WS) specifications offers functionalities useful enough as to create awareness models in an environment. In the same way, none of the last WS specifications offers specific functionalities to manage different levels of awareness in cooperative environments.

4 CAM's Semantic Web implementation

As it has been exposed in previous sections CAM model is a specific approach to create VOs in Grid environments. This paper complements the research carried out around CAM with the development of an ontology for the CAM model. This ontology will bring meaningful knowledge in CAM's Grid environments, because it specifies the concepts of CAM in an unambiguous and machine-understandable way. This process has been mainly done by composing an ontology which contains CAM's key concepts allowing reasoning about specific situations in the Grid environment.

The domain of CAM ontology is the representation of CAM's concepts like *Environment*, *Resource*, *Task*, and *InteractivePool*. The ontology will be used to determine the relations between the resources based on the properties *Aura*, *Focus* and *Nimbus*. The final aim of the ontology is to determine the awareness of interaction between the resources and therewith to determine e.g. their membership in different *InteractivePool* in order to carry out a *TaskResolution*. The membership to a specific *CooperativeOrganization* will be defined by means of applying reasoning to CAM's concepts. The gap between *InteractivePool* and the final *TaskResolution* is fulfilled with SPARQL [1] queries that pull out the needed score to execute a given process in a specific resource.

Therefore CAM is implemented with SW technologies with the following component structure:

- CAM's concepts in terms of ontology classes and its attributes.
- Relationships between CAM classes.
- Rules to infer the derivated classes *AwarenessInteraction*, *InteractivePool* and *CooperativeOrganization*. As it will shown later this rules are directly translated from logic formulas developed in section 3.
- Queries in SPARQL format that extracts the scores to perform processes in nodes. *TaskResolution* concept is the outcome of running this logic.

By building this architecture which plugs together a rule engine, a reasoner and a triple store that can be semantically queried, it is possible to infer e.g. potential memberships in VO or, regarding the indirect awareness, Cooperative Organizations and any other concept that potentially could extends CAM's ontology.

4.1 Bits of the ontology

Figure 1 shows CAM's classes and its relationships:

- Resource: Represents the Grid key concept Resource (an entity that is useful in the Grid environment, like CPU power, disk space or a certain application)
- ComputationElement: Represents the underlying hard and software of a Resource. It is modeled in an extra class to keep the Resource class clearly defined
- VirtualOrganization: Represents the concept Virtual Organization.
- Environment: Represents a distributed grid environment that contains a set of resources.
- Aura: Represents the CAM concept Aura.
- Task: Represents a job and its requirements needed to be solved

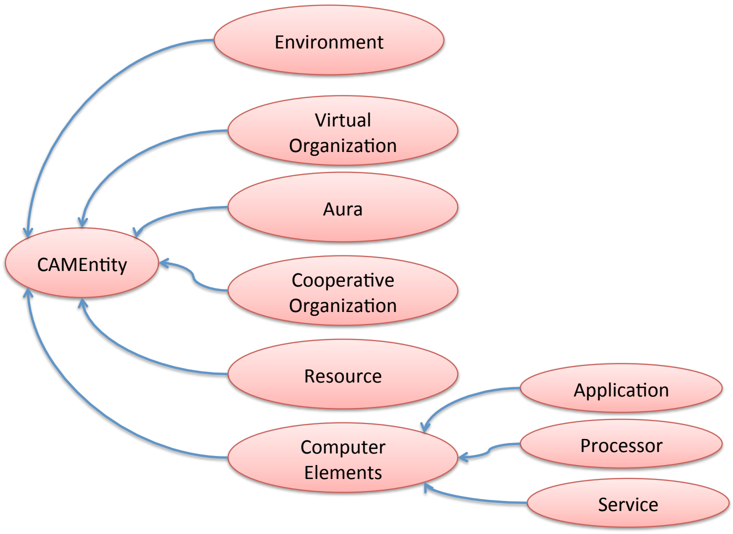

Fig. 1. The CAM ontology classes

The relationships between the different CAM classes are defined by the properties enumerated in tables 1 and 2. For the shake of simplicity just the concepts *Environment* and *Resource* along with their properties and attributes are described :

Class: **Environment**		
Property	**Range**	**Characteristics**
contains	Resource	inverseOf belongsTo
boundedByAura	Aura	-
containsBounded	Resource	-

Table 1

The *Enviroment* concept exposes the *contains* attribute applied to *Resource*, actually an instance of *Enviroment* makes sense as set of resources that compounds it. *boundedByAura* property connects the environment with aura that delimits it and *containsBounded* gives the subset of resources bounded by the aura.

Table 2 gives an overview of all object properties under *Resource* class. A subset, the one consider remarkable, is shortly explained in the following:

- belongsTo Every resource belongs to an environment.
- hasFocusOn Relates to the CAM concept *focus*. A resource could have its focus on other resources.
- hasNimbusSpaceOn The CAM concept *nimbus* is a tuple containing NimbusSpace and NimbusState. In this ontology, nimbus is split up in two according properties. A resource could have its nimbus space on other re-

Class: **Resource**		
Property	**Range**	**Characteristics**
belongsTo	Environment	inverseOf contains
hasFocusOn	Resource	inverseOf isInFocusOf
inInFocusOf	Resource	inverseOf hasFocusOn
hasNimbusSpaceOn	Resource	inverseOf isInNimbusSpaceOf
isInNimbusSpaceOf	Resource	inverseOf hasNimbusSpaceOn
hasAwareIntUniFullOf	Resource	-
hasAwareIntBiFullOf	Resource	-
hasInteractivePool	Resource	-
ssignedToTask	Task	functional
hasComputerSystem	ComputerSystem	functional
hasVirtualOrganization	VirtualOrganization	inverseFunctional isVirtualOrganization
hasNimbusState	Collection	{Null, Medium, Maximum}

Table 2

sources.

- hasNimbusState The state of the system, depending on the system load. The three possible values {Null, Medium, Maximum} are mapped in this ontology as set of constants.
- hasAwareIntUniFullOf Represents the CAM concept of unidirectional *Awareness of Interaction* between a pair of resources, this arch is derived with rule 1.
- hasAwareIntUniFullOf Represents the CAM concept of bidirectional *Awareness of Interaction* between a pair of resources, this arch is derived with rule 2.

4.2 The inference rules

As it was in previous sections, together with the classes are needed a set of rules in order to infer an *InteractivePool*. The rules defined from now make use of two namespaces *cam* and *rdf*, they respectively refers to CAM's and RDF's namespaces. The grammar of the rules is a triple based pattern used within Jena [7] framework. The CAM's semantic web infrastructure makes the most of Jena developing the reasoning functionalities on top of it .

The type of reasoning used is data driven, also known as forward chaining.

Data driven reasoning takes as input the data and together with the rules extracts or infer more data, also named derived or inferred data. In order to get *InteractivePool* objects three rules are attached to the reasoner to be triggered on cascade when any of the inferred concepts are queried:

Rule 1 raises the arch *cam:hasAwareIntUniFullOf* between two different *cam:Resource* instances.

```
(?x rdf:type cam:Resource) (?y rdf:type cam:Resource)
notEqual(?x,?y)
(?x cam:hasFocusOn ?y) (?y cam:hasNimbusSpaceOn ?x)
-> (?x cam:hasAwareIntUniFullOf ?y )
```

Rule 2 creates the arch *cam:hasAwareIntBiFullOf* between two different *cam:Resource* instances that are linked through the arch *cam:hasAwareIntUniFullOf* triggered by rule 1.

```
(?x rdf:type cam:Resource) (?y rdf:type cam:Resource)
notEqual(?x,?y)
(?x cam:hasAwareIntUniFullOf ?y)
(?y cam:hasAwareIntUniFullOf ?x)
-> (?x cam:hasAwareIntBiFullOf ?y )
```

Rule 3 sets the collection *InteractivePool* through the set of arches *cam:hasInteractivePool* that links two resources with bidirectional *Awareness of Interaction*, definided by *cam:hasAwareIntBiFullOf* in rule 2.

```
(?x rdf:type cam:Resource) (?y rdf:type cam:Resource)
notEqual(?x,?y)
(?y cam:hasAwareIntUniFullOf ?x)
-> (?x cam:hasInteractivePool ?y )
```

4.3 The SPARQL queries

To this point the ontology structure and rules have been defined, the piece left to describe is the SPARQL queries. SPARQL is a query language standardized by the W3C. SPARQL defines a query syntax to RDF graphs relying on triple pattern definitions. The results of a SPARQL query can be either a result set in XML format or an RDF.

The queries to pull out the information that build the *TaskResolution* are created dynamically, therefore an instance of one those queries created by the middleware is shown below:

```
1      PREFIX cam: <http://www.ecs.soton.ac.uk/~ms8/cam#>
2      PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3      SELECT ?node ?serviceId ?score WHERE {
4      { cam:Node1 cam:hasInteractivePool ?node .
6        ?node cam:hasNimbusState ?nimbusState .
7        ?node cam:hasComputationElement ?computational .
8        ?computational cam:hasName ?serviceId .
9        ?computational cam:hasScore ?score .
10        FILTER(
11           (?serviceId = cam:Maximum || ?nimbusState = cam:Medium)
12               &&
13           regex(?serviceId,"ServiceA|ServiceB"))
15      } }
```

This query collects three pieces of information represented by the nodes variables *node*, *serviceId* and *score*. Therefore each collected record contains the node identification where a specific service could it be executed with a given score. In the query, from line 4 to 9, are established the triple patterns to constraint the information collected. From line 10 to 15, are defined the filter conditions. In the example shown the filter conditions leave to go through all the nodes either with *Maximum* or *Medium nimbusState* and that contains some service named "ServiceA" or "ServiceB".

4.4 Putting all together

This section, so far, has developed the three SW components used by CAM. The architecture in terms of how this elements are connected between them is shown in figure 2.

Each computational node, to be connected into CAM infrastructure needs exposing an RDF with information according to CAM's ontology. The node which receives the request to carry out a task is in charge of gathering all the RDF files with information within its *Enviroment*. If the node belongs to more than *Enviroment* then it should be specified under which one the task is requested to be executed.

Once that all the information is gathered into a single RDF graph then that graph together with the inference rules are used to set the reasoner up. The next step implies the execution of the SPARQL query which is the element that activates the reasoner since the inferred property *cam:hasInteractivePool* is part of the query (see section 4.2).

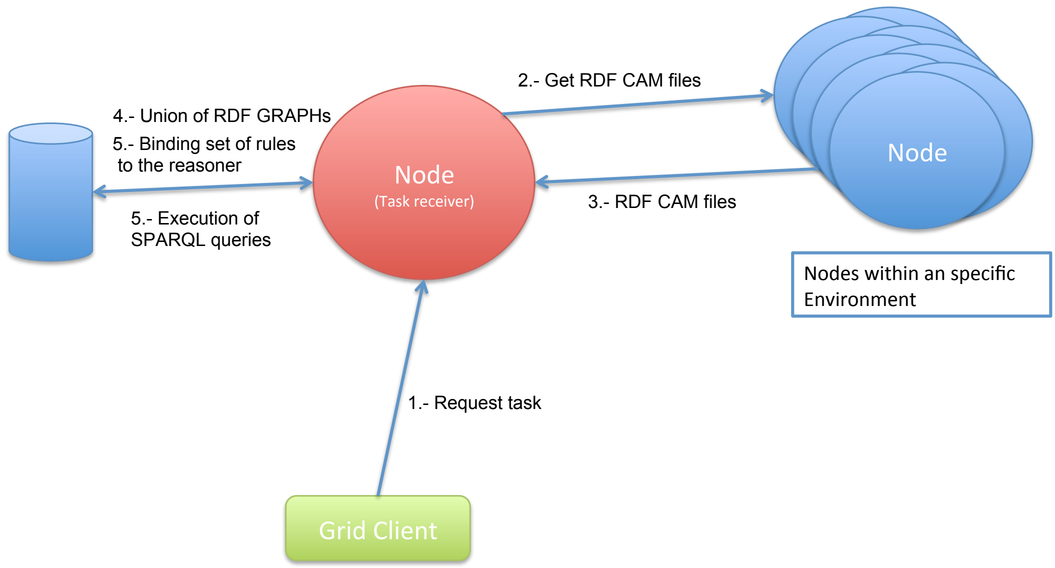

Fig. 2. Interaction between the components

5 Conclusions and future work

Managing configurations to share resources is complex more even if the scenario is a Grid Computing distributed system. Currently Grid Computing middlewares lack of a way to create VOs. CAM has come up in order to fill that gap. Along this paper CAM model has been explained in-depth as well as its ontology representation.

Defining the ontology for CAM model allows us to define semantic management information which move us one step further, into the Semantic Grid. Semantic Web will allow CAM to perform reasoning over the grid management data as well as mediation between Grid services in third party models. These research advances will be presented in future work.

References

[1] Sparql query language for rdf http://www.w3.org/tr/rdf-sparql-query/ (accesed on 04/2008).

[2] Open grid forum, www.ogf.org, August 2007.

[3] Standards development organizations collaboration on networked resources management (scrm-wg), August 2007.

[4] Liam J. Bannon and Kjeld Schmidt. Cscw: Four characters in search of a context. pages 3–16, 1991.

[5] Steve Benford and Lennart E. Fahlen. A spatial model of interaction in large virtual environments. In *ECSCW*, pages 109–124, 1993.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *The Scientific American*, May, 2001.

[7] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. Technical Report HPL-2003., 2003.

[8] Oscar Corcho, Pinar Alper, Ioannis Kotsiopoulos, Paolo Missier, Sean Bechhofer, and Carole Goble. An overview of S-OGSA: A Reference Semantic Grid Architecture. *Journal of Web Semantics*, March 2006. www.cs.man.ac.uk/~penpecip/pblctns/S-OGSA-JWS.pdf consulted in August 2007.

[9] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiologyyyyyyy of the grid: An open grid services architecture for distributed systems integration, 2002.

[10] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 2003.

[11] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:25, 2001.

[12] C. Greenhalgh. *Large Scale Collaborative Virtual Environments*. PhD thesis, University of Nottingham, 1997.

[13] Pilar Herrero, José Luis Bosque, Manuel Salvadores, and María S. Pérez. Amble: An awareness model for balancing the load in collaborative grid environments. In *The 7th IEEE/ACM International Conference on Grid Computing (Grid 2006)*, 2006.

[14] Pilar Herrero, José Luis Bosque, Manuel Salvadores, and María S. Pérez. A collaborative awareness specification to cover load balancing delivery in cscw grid applications. *The Second International Workshop on Agents, Web Services and Ontologies (AWeSOMe'06)*, pages 78–89, 2006.

[15] Pilar Herrero, José Luis Bosque, Manuel Salvadores, and María S. Pérez. Managing Dynamic Virtual Organizations to get Effective Cooperation in Collaborative Grid Environments. In *The Second International Conference on Grid computing, High-Performance and Distributed Applications (GADA '07) in conjunction with On The Move Federated Conferences 2007 (OTM '07)*, Vilamoura, Algarve, Portugal, November 2007.

[16] Pilar Herrero, José Luis Bosque, Manuel Salvadores, and María S. Pérez. A rule based resources management for collaborative grid environments. In *International Journal of Internet Protocol Technology*, 2007.