Managing Dynamic Virtual Organizations to Get Effective Cooperation in Collaborative Grid Environments

Pilar Herrero[1], José Luis Bosque[2], Manuel Salvadores[1], and María S. Pérez[1]

[1] Facultad de Informática
Universidad Politécnica de Madrid
Madrid. Spain
{pherrero, mperez}@fi.upm.es
[2] Dpto. de Electrónica y Computadores
Universidad de Cantabria
Santander, Spain
joseluis.bosque@unican.es

Abstract. This paper presents how to manage Virtual Organizations to enable efficient collaboration and/or cooperation as a result of a flexible and parametrical model. The CAM (Collaborative/Cooperative Awareness Management) model promotes collaboration around resources-sharing infrastructures, endorsing interaction by means of a set of rules. This model focuses on responding to specific demanding circumstances at a given moment, while optimizes resources communication and behavioural agility to get a common goal: the establishment of collaborative dynamic virtual organizations. This paper also describes how CAM works in some specific examples and scenarios, and how the CAM Rules-Based Management Application (based on Web Services and named WS-CAM) has been designed and validated to encourage resources to be involved in collaborative performances, tackling efficiently demanding situations without hindering the own purposes of each of these resources.

1 Introduction

The rapid evolution of information and the new potentials for team work have been of great importance to the success of most complex organizations. Activities in the domain of computer support for team work are well-known, since last decade, by the notions of groupware or Computer-Supported Cooperative Work (CSCW). In fact, Ellis [12] defines groupware as "computer-based systems that support groups of people engaged in a common task and that provide an interface to a shared environment." while according to Wilson [23] is "CSCW a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques."

R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part II, LNCS 4804, pp. 1435–1452, 2007.

Computer supported cooperative work (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. The CSCW paradigm has traditionally encompassed distributed systems technologies such as middleware, business process management and web technologies. However new tendencies – such as Grid Computing - have modified the technological scenery of this kind of systems. Applications in this kind of systems are highly distributed and coordinated; exhibiting different patterns of interaction and requiring distributed access and sharing multiple heterogeneous resources to be able to afford highly performance computing problems by virtual computer architecture. The objective is to obtain a flexible, secure and coordinated resource sharing organization among dynamic collections of resources in a dynamic, stable, flexible and scalable network.

Sharing is a very broad concept that could be used in different concepts with very different meanings. In this research we understand sharing as: "*access to resources, as is required by a range of collaborative problem-solving and resource brokering strategies"*, and, according to Ian Foster [13], the set of individuals and/or institutions defined by such sharing rules form is called V*irtual Organization* (VO) in grids.

However, which kind of "sharing" rules should be applied?, when?, and why? Which are going to be the conditions to share theses resources? In short, something that is still missing but needed in this kind of systems: how to manage resources according to set of rules. These rules, defined by each of the component of this dynamic infrastructure, will allow having "management policies" for open distributed systems. Having a look to all these questions, the key issue to be achieved in order to manage collaborative dynamic virtual organizations, seems to be the definition of "sharing" rules, and then, the question to be raised is how to define these rules. This paper focuses on the management of resources by means of Collaborative/ Cooperative Awareness Management (CAM) model and its implementation (WS-CAM).

CAM has been designed, form the beginning to be a parametrical, generic, open, model that could be extended and adapted easily to new ideas and purposes. 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. This model has also been designed to apply successful agent-based theories, techniques and principles to deal with resources sharing as well as resources assignment inside the environment. As for the implementation, WS-CAM, it has been made to be generic, open, easy to be extended, adaptable to new modifications in the model, scalable and free of bottleneck

CSCW in grid infrastructures and business rules are natural allies as they can benefit mutually. The use of business rules gives business agility in terms of being able to change the way the systems responds when circumstances demand it. CSCW in grid environments allow managing large number of heterogeneous resources while they cooperate, by means of these rules, to accomplish a common goal. A combined system, as the one presented in this paper, offers cooperation and behavioural agility.

2 Related Work

Managing resources in a large scale environment involves critical aspects, such as scheduling [22], discovery [5], load balancing [4] or fault tolerance [16]. Several approaches have addressed all these topics. However, as far as we know, there are not any awareness-based systems used to deal with the management of resources. In [3] presents an example, in an e-Government scenario in which public administrations cooperate in order to fulfil service requests from citizens and enterprises. Service-based Cooperative Information Systems's consider cooperation among different organizations to be obtained by sharing and integrating services across networks such as e-Services and Web-Services [20]. In the literature, CIS's have been widely considered, [9], [24]; various approaches are proposed for their design and development: schema and data integration techniques [7], agent-based methodologies and systems [19], and business process coordination and service-based systems [10].

Business rules or business rulesets could be defined as a set of "operations, definitions and constraints that apply to an organization in achieving its goals" [8]. Business rules produce knowledge. They can detect that a situation has occurred and raise a business event or even create higher level business knowledge. Business rules engines help manage and automate business rules, registering and classifying all of them; verifying their consistency and even inferring new rules. From all the possible Rules Engines, [8], we have selected JBoss Rules (Drools) for achieving our purposes. Drools is a Rule Engine but it is more correctly classified as a Production Rule System, a kind of Rule Engine and also Expert System [18].

3 CAM: Collaborative/Cooperative Awareness Management

CAM, which allows managing awareness in cooperative distributed systems, has been designed based on the extension and reinterpretation of the Spatial Model of Interaction (SMI) [6], an awareness model designed for Computer Supported Cooperative Work (CSCW). This reinterpretation, open and flexible enough, merges all the OGSA [14] features with theoretical principles and theories of multi-agents systems, to create a collaborative and cooperative environment within which it is possible to manage different levels of awareness.

Given an distributed environment (E) containing a set of resources $E=\{R_i\}$, and a T task which needs to be solved in this environment, if this task is made up by a set of tuples (p_i, rq_i):

$$T= \sum(p_i, rq_i),$$

Where "p_i" are the processes needed to solve the T task in the system. These processes could be related to power, disk space, data and/or applications. And "rq_i" are requirements needed to solve each of these p_i processes, such as power, disk space, data and/or applications. CAM intends to solve the T task in a collaborative and, if possible, cooperative way, by extending and reinterpreting the SMI's key concepts in the context of Grid Environments:

- *Focus*: It can be interpreted as 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. Given a resource in the system, its focus would contain, at least, the subset of resources that are composing the Virtual Organization (VO) [21] in which this resource is involved but it could be modified and oriented towards any other VO, if needed.

- *Nimbus:* It is defined as a tuple (Nimbus=(*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 properties, such as load of the system, disk space, data information stored, 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/cooperative process.

- *Awareness of Interaction (AwareIntRi→Rj):* Probably the best-known definition of awareness in CSCW literature was given by Dourish et al [11] in their paper on awareness and co-ordination in shared workspaces. They define awareness as "an understanding of the activities of others which provides a context for your own activity". This concept will quantify the degree, nature or quality of asynchronous unidirectional interaction between a pair of distributed resources in the Environment (E). Following the awareness classification introduced by Greenhalgh in [25], this awareness could be *Full*, *Peripheral* or *Null*.

$$AwareInt_{R_i \to R_j}(E) = \begin{cases} R_j \in Focus(R_i) \ \wedge \ R_i \in Nimbus(R_j) & Full \\ (R_i \in Focus(R_j) \ \wedge \ R_j \notin Nimbus(R_i)) & \\ \vee & Peripheral \\ (R_j \notin Focus(R_i) \ \wedge \ R_i \in Nimbus(R_j)) & \\ Otherwise & Null \end{cases}$$

- *Awareness of Interaction (AwareInt)*: This concept will quantify the degree, nature or quality of asynchronous bidirectional interaction between a pair of distributed resources. This awareness could also be *Full*, *Peripheral* or *Null*.

$$AwareInt(R_i, R_j) = \begin{cases} AwareInt_{R_i \to R_j}(E) = Full \ \wedge \ AwareInt_{Rj \to Ri}(E) = Full & Full \\ AwareInt_{R_i \to R_j}(E) = Full \ \oplus \ AwareInt_{Rj \to Ri}(E) = Full & Peripheral \\ Otherwise & Null \end{cases}$$

- *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.
- *Interactive Pool:* This function returns the set of resources interacting with a given resource in a given moment.

$$If\ AwareInt_{A \to B}(A, B) = Full\ then\ B \in InteractivePool\ (A)$$

- *Task Resolution*: This function determines if there is a service in the resource B, being NimbusState (B)/=Null, such that could be useful to execute T (or at least a part of this task).

$$TaskReolution\ (B, T) = \{(p_i, s)\}$$

Where *s* is the "score" to carry out p_i in the B resource, being its value within the range [0, ∞): 0 if the B resource fulfils all the minimum requirements to carry out the process p_i; the higher is the surplus over these requirements, the higher will be the value of this score. This concept would also complement the Nimbus concept, because the *NimbusSpace* only determines those machines that could be taking into account in the assignment process because they are not overloaded yet. However, the *Task Resolution* determines which of these machines can contribute effectively to solve T or, at least, a part of this task.

- *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.

$$If\ AwareInt_{A \to B}(A, B) = Full\ then\ B \in InteractivePoll\ (A)$$

$$If\ TaskResolution\ (B, T) = \{(p_i, s)\}\ then\ B \in VirtualOrganization\ (A, T)$$

Resources belonging to this VO could access to resources, as they are aware of them, to solve specific problems, and they could collaborate each other, getting therefore a Virtual Organization (VO) [13]. 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 [2].

CAM also intends to determine if resources cooperate among them in the context of Grid Environments by means of the following concepts:

- *Cooperative Awareness of Interaction (*$CoopAwareInt_{Ri \to Rk \to Rj}$*)*: This concept will quantify the degree, nature or quality of asynchronous interaction between distributed resources. In Computer Supported Cooperative Work (CSCW), this awareness could be due to the direct or indirect interaction between resources [15]. In fact, the awareness that a resource (R_i) has of another one (R_j) could be associated to the presence of a third resource (R_k) in the environment. In this way, if a resource (R_1) is aware of another resource (R_3) and this resource (R_3) is aware of another one (R_2), then R_2 could be an "*indirect"* aware of the first one by means of R_3 (see Figure 1).

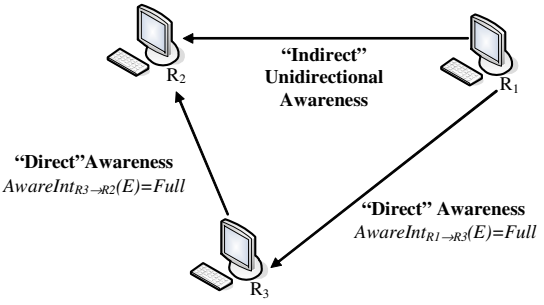

Fig. 1. A Scheme of unidirectional "Indirect" Awareness in CSCW

This "indirect" awareness could also be unidirectional (Figure 1) or bidirectional (Figure 2).

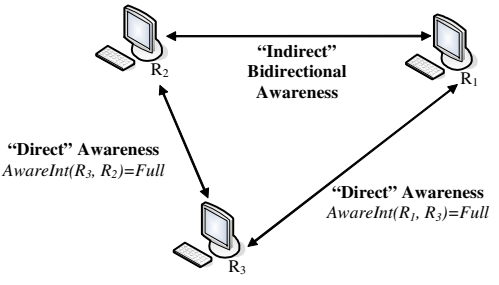

Fig. 2. A Scheme of unidirectional "Indirect" Awareness in CSCW

In CSCW, the awareness that a resource has of an item one could also be distorted by the presence of an additional item. In this way, let's suppose a medium where a resource (R_1) is aware of the R_2 resource. If while R_1 is being aware of R_2 a third item (R_3) appears, this "new" could distort the interaction (and therefore) the awareness in a positive or negative way (Figure 3).

Taking into account the previous situations, the cooperative awareness of interaction ($CoopAwareInt_{Ri\rightarrow Rk\rightarrow Rj}$) is defined as a tuple (*TypeAwareness, TypeInteraction, StateAwareness*) containing information about:

- The type of the awareness of interaction (*TypeAwareness*): *Indirect* or *Distorted*.
- The type of interaction (*TypeInteraction*): *Unidirectional* or *Bidirectional*.
- The state of the awareness of interaction after this cooperation (*StateAwareness*): *Full*, *Peripheral* or *Null*.

- *Cooperative Directional Pool:* This function returns the set of resources cooperating among them, uni-directionally, in a given moment (StateAwarenes=Full).

If $CoopAwareInt_{Ri\rightarrow Rk\rightarrow Rj}$(E)=(Indirect, Unidirectional, Full) then

$$CooperativePool_{Ri\rightarrow Rj}(E)=\{ R_i, R_k, R_j\}$$

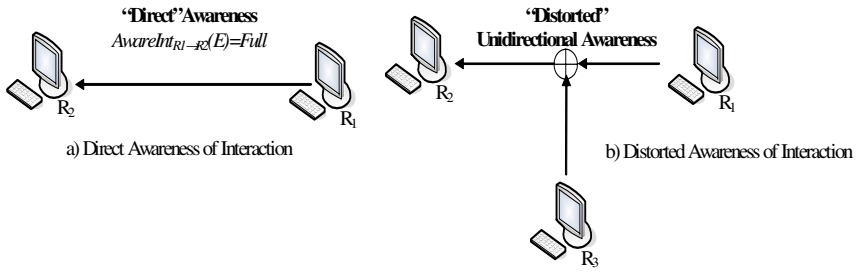

Fig. 3. "Distorted" Awareness in CSCW

- *Cooperative Pool:* This function returns the set of resources cooperating among them, bi-directionally, in a given moment (StateAwarenes=Full).

  If $CoopAwareInt_{Ri \to Rk \to Rj}(E)$=(Indirect, Bidirectional, Full) then

  $CooperativePool(E)=\{ R_i, R_k, R_j\}$

- *Cooperative Directional Organization:* This organization will be made up by the set of resources cooperating, uni-directionally, in the environment.

  $CoopAwareInt_{R1 \to R2 \to R3}(E)$=(Indirect, Unidirectional, Full) =>

  $CooperativePool_{R1 \to R3}(E)=\{ R_1, R_2, R_3\}$

  $CoopAwareInt_{R3 \to R4 \to R5}(E)$=(Indirect, Unidirectional, Full) =>

  $CooperativePool_{R3 \to R5}(E)=\{ R_3, R_4, R_5\}$

  $CooperativeOrganization_{R1 \to R5}(E)=\{ R_1, R_2, R_3, R_4, R_5\}$

- *Cooperative Organization:* This organization will be made up by the set of resources cooperating, bi-directionally, in the environment.

  $CoopAwareInt_{R1 \to R2 \to R3}(E)$=(Indirect, Bidirectional, Full) =>

  $CooperativePool(E)= \{ R_1, R_2, R_3\}$

  $CoopAwareInt_{R3 \to R4 \to R5}(E)$=(Indirect, Bidirectional, Full) =>

  $CooperativePool(E)=\{ R_3, R_4, R_5\}$

  $CooperativeOrganization(E)=\{ R_1, R_2, R_3, R_4, R_5\}$

As far as we know, none of the last 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 Rules-Based Management: Autonomic Computing

Let's consider a business organization made up by several departments and sub-departments in which there are different local rules to use resources. These rules are executed by the information technology department. As the guidelines for sharing resources need to be modified, continuously, with the time and they also need to lead an indeterministic system, such as CAM, due to the decentralised broker it provides, the complexity of the problem is high. This complexity is even bigger as we should also consider the complexity of managing a distributed system, and even more due to the growth of functional requirements, quality of services and the increase of heterogeneous resources.

Basically, due to the rapidly growing complexity, to enable their further growth, IBM started in 2001 the Autonomic Computing initiative [17], which aims of creating self-managing computer systems. Autonomic Computing defines four functional areas: Self-Configuration (Automatic configuration of components); Self-Healing (Automatic discovery, and correction of faults); Self-Optimization (Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements); and Self-Protection (Proactive identification and protection from arbitrary attacks). From these four functionalities, WS-CAM endows collaborative/cooperative environments with three of them, Self-Configuration, Self-Optimization and Self-Healing to work properly without human intervention.

WS-CAM Self-configuration:

- Self-deployment of services: From all the resources available in the environment it can determine which resources can offer specific services as well as the service's auto-deployment. It is also possible to include temporal advantages based on some planning rules to deploy and remove specific services.
- Self-awareness: A resource can be aware of those resources, in its surrounding, which could be useful to carry out a specific task, and vice-versa, it can also be aware of those resources for which it could be useful.
- Self-parameter configuration: A resource could be able to modify any of its parameters based on a set of rules. This service could also be applied to the modification of the key concepts of WS-CAM.

WS-CAM Self-Optimization:

- Self-parameter optimization: WS-CAM can determine which resource is more suitable not just to carry out a task (such as "access to a Data Base") but also a Collaborative/Cooperative task. Once the resource has been identified, it will also deploy the corresponding service.

WS-CAM Self-Healing:

- Self-Proactive discovery: If several resources are collaborating/cooperating and the "Self-parameter optimization" service detects that a resource (A) is getting overloaded, the "self-Proactive discovery" service will determine, from all the resources available in the collaborative organization, which of them would be more suitable to carry out, if possible, some of the processes that A is developing, deploying automatically the corresponding services, reducing the A's load and ensuring the optimal work of the system.

These three functionalities of WS-CAM intend to reduce the effort associated to the complexity of the system, reducing responding and recovering time in very dynamic environments and scenarios, as the presented in this paper.

5 WS-CAM Rules-Based Management Architecture

As it is possible to appreciate in the figure (Figure 4):

- Each of the resources publishes its computational and administrative properties and attributes (WS-DataExtension).

- The resources involved in this management will be able to receive information about how these parameters (properties and attributes) have been defined once it has been published. In the same way, they will receive information about their corresponding modifications.
- CAM will keep a historic with all those collaborations that were successfully carried out in the environment with the aim of being taking into account for future collaborative/cooperative task – selecting automatically, the resource more suitable to carry out a single task and/or the resource more suitable for a Collaborative/Cooperative task -as well as for optimising purposes.
- The ***optimization agent*** will check the information stored in the "historic" of the system, analysing the role that each of the resources involved in a collaborative task played in task's resolution. An example of this optimization could be, i.e. "*if B∈ Focus (A) but A didn't use the B's services in past 80% collaborative task, then remove B from the A's Focus*". This optimization could be automatic or semi- automatic.
- The ***configuration agent*** will be endowed with a set of rules to trigger the corresponding modifications in the system. These rules could be classified as configuration and optimization rules. An example of configuration rules could be: "*All the nodes belonging to the manufacturing department with, at least, 2GB of memory will be available for the whole distributed environment from Monday to Friday, form 1 to 8 am"*. This rule modifies the Focus, the Nimbus and therefore the Awareness of the system

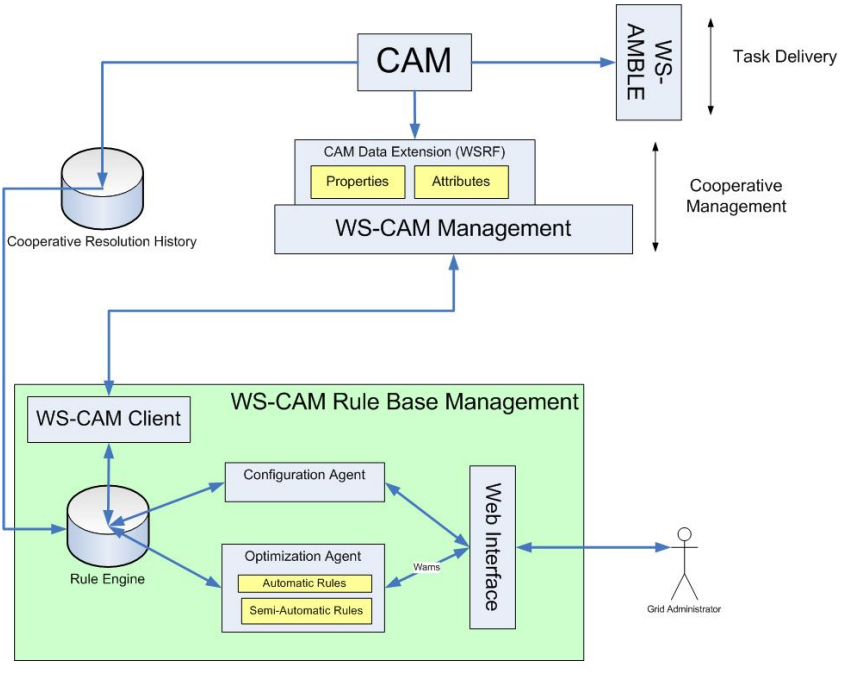

Fig. 4. WS-CAM Management Architecture

6 CAM's Validation

The aim of this section is to validate the cooperative model described throughout this paper. For this purpose, we designed a process of evaluation consisting of three different steps: scenario-based validation, user-based validation and the last one performance-based validation.

6.1 Scenario-Based Validation

Let' assume an Environment (E) bounded by the a_1 aura (E_{a1}), where E_{a1} = {R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8}, let's assume that the R5 resource intends to solve the T task, T = {(p_1,rq_1),(p_2,rq_2),(p_3, ,rq_3),(p_4, ,rq_4)} by using the following rule:

Name: Focus Biological Department
Type: addFocus
Operation: S0= Linux & RAM=2GB

This rule will include, automatically, in the R_5's Focus all those resources fulfilling these requirements: Focus(R_5) ={ R_2, R_3 }

Initially, the NimbusSpace of each of these nodes is:

NimbusSpace (R_1)={ } NimbusSpace (R_2)={ R_1, R_2, R_3, R_5}
NimbusSpace (R_3)={ R_3, R_7} NimbusSpace (R_4)={ }
NimbusSpace (R_6)={ } NimbusSpace (R_7)={ }
NimbusSpace (R_5)={ R_1, R_2, R_3, R_4, R_5, R_6, R_7}

Let's also consider three powerful resources (R_4, R_5 & R_7) are switched on in the system with the rule:

Name: Nimbus Proteins Department
Type: addNimbus
Operation: Processor= DualCore | Speed=2GHz & user=Root

Being therefore:
NimbusSpace(R_4)= NimbusSpace(R_6)=NimbusSpace(R_7)={R_1, R_2, R_3, R_4, R_5, R_6, R_7}
If the R_7's resource has an additional rule:

name: "PowerProjection: Time Constraint"
type: "Nimbus"
Operation: "Time restriction: Monday, form 5 am to 8 pm"

Getting therefore:
If ((5:00 ≤ time ≤ 20:00) ^ (day=Monday)) then
{NimbusSpace(R_7)= { R_1, R_2, R_3, R_4, R_5, R_6, R_7}}
Else {NimbusSpace(R_7)={ }}

Let's also consider T={p1, p2, p3, p4} wants to carry out the T task, which also requires some specific disk space properties to manage the data information, the *NimbusState* of each of this resources could have three possible values (*Null*, *Medium* or *Maximum)* depending on the resource's properties

NimbusState (R_1,T)= NimbusSpace (R_3,T)= NimbusSpace (R_5,T)=Null

NimbusState (R_2,T)= NimbusState (R_6,T)=Medium
NimbusState (R_4,T)= NimbusSpace (R_7,T)= Maximum

Combining NimbusState and NimbusSpace, the Nimbus these resources will be:

Nimbus (R_1,T)=(Null,{ })
Nimbus (R_2,T)=(Medium, { R_1, R_2, R_3, R_5})
Nimbus (R_3,T)=(Null, { R_3, R_1})
Nimbus (R_4,T)=(Maximum,{ R_1, R_2, R_3, R_4, R_5, R_6, R_7})
Nimbus (R_5, T)=(Null, { R_1, R_2, R_3, R_4, R_5, R_6, R_7})
Nimbus (R6, T)=(Medium, { R_1, R_2, R_3, R_4, R_5, R_6, R_7})
Nimbus (R_7, T)=(Maximum, { R_1, R_2, R_3, R_4, R_5, R_6, R_7})

Taking into account the previous values, as well as Focus(R_5) ={ R_2, R_3}, the CAM will calculate the awareness of interaction among them:

$AwareInt_{R5 \rightarrow R2}(E_{a1})$= Full
$AwareInt_{R5 \rightarrow R3}(E_{a1})$ = Peripheral
$AwareInt_{R5 \rightarrow R4}(E_{a1})$ =$AwareInt_{R5 \rightarrow R6}$ (E_{a1}) = $AwareInt_{R5 \rightarrow R7}$ (E_{a1})=Null

Only those resources whose awareness of interaction with R_5 was Full will be part of the Interactive Pool of R_5. InteractivePool(R_5)= {R_2}

As their nimbus state is: NimbusState(R_2)= Medium

If the task resolution is: TaskResolution(R_2,T)={ (p_1,1), (p_2,0.8)}

Then VirtualOrganization(R_5,T)= {R_2} will not be able to solve the T task, and the rules engine will automatically increase the aura ($a_1 \rightarrow a_2$), modifying the focus, the nimbus (NimbusSpace) and therefore the awareness.

If the System with this new aura includes two new nodes (R_9 & R_{10}), Ea_2 = {R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}}, and the R_5's Focus increases to: Focus(R_5) ={R_2, R_3, R_9, R_{10}}

In the same way, there is a rule for R_9 & R_{10} resources such as:

name: "PowerProjection: 15 processors, 7GB with Time Constraint"
type: "Nimbus"
Operation: "Project its characteristics from Monday to Friday, form 1 to 8 am"

Getting therefore:

If ((1:00 ≤ time ≤ 8:00) ^ (Monday ≤ day ≤ Friday))
{NimbusSpace(R_9)={R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_9, R_{10}}}
Else {NimbusSpace(R_9)={ }}

Let's also assume we keep working with the same task, T={ p_1, p_2, p_3, p_4}. If:
NimbusState (R_9,T)= NimbusSpace (R_{10},T)= Maximum
Nimbus (R_9, T)= Nimbus (R_{10}, T) = (Maximum, { R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_9, R_{10} })

Taking into account the previous values, as well as Focus(R_5) ={R_2,R_3, R_9, R_{10}}, the CAM will calculate the awareness of interaction among them:

$AwareInt_{R5 \rightarrow R2}(Ea_2)$= $AwareInt_{R5 \rightarrow R9}(Ea_2)$= $AwareInt_{R5 \rightarrow R10}(Ea_2)$= Full

Only those resources whose awareness of interaction with R5 was Full will be part of the Interactive Pool of R_5: InteractivePool(R_5)= {R_2, R_9, R_{10}}

As their nimbus state is: NimbusState(R_2) = Medium, NimbusState(R_9)= NimbusState(R_{10})= Maximum and the task resolution is:

TaskResolution(R_2,T)={ (p_1,1), (p_2,0.8)}
TaskResolution(R_9,T)={ (p_2,0.8), (p_3,1), (p_4,0.5)}
TaskResolution(R_{10},T)= {(p_1,1) }

We can conclude that: VirtualOrganization(R_5,T)= {R_2, R_9, R_{10}} can solve successfully the T task with no problem.

Moreover, if the following rule is considered:

name: "Disk Space Selection"
type: "Focus"
Operation: "Disk Space ≥ 10GB"

This rule will include, automatically, in the R_2's Focus all those resources fulfilling these requirements:

Focus(R_2) ={ R_{10} }
NimbusSpace (R_{10})={R_1,R_2 }
$AwareInt_{R2 \rightarrow R10}(E_{a2})$= Full

As it was mentioned before: $AwareInt_{R5 \rightarrow R2}(E_{a2})$= Full

Having therefore: $CoopAwareInt_{R5 \rightarrow R2 \rightarrow R10}(E_{a2})$ = (Indirect, Unidirectional, Full)
Getting a cooperative pool: $CooperativePool_{R5 \rightarrow R10}(E_{a2})$={ R_5, R_2, R_{10}}
On the other hand, a new rule could be introduced for the R_{10} resource:

name: "CPU-Power Selection"
type: "Focus"
Operation: "CPU-Power ≥ 2GHz"

This rule will include, automatically, in the R_{10}'s Focus all those resources fulfilling these requirements:

Focus(R_{10}) ={ R_8 }
NimbusSpace (R_8)={R_1,R_{10} }
$AwareInt_{R10 \rightarrow R8}(E_{a2})$= Full

As it was mentioned before: $AwareInt_{R2 \rightarrow R10}(E_{a2})$= Full
Having therefore: $CoopAwareInt_{R2 \rightarrow R10 \rightarrow R8}(E_{a2})$ = (Indirect, Unidirectional, Full)
Getting a cooperative pool: $CooperativePool_{R2 \rightarrow R8}$ (E)={ R_2, R_{10}, R_8} then

$$CooperativeOrganization\ (E) = \{ R_5, R_2, R_{10}, R_8\}$$

6.2 User-Based Validation

This part of the validation process consists of a scenario made up by a number of tasks that users need to complete to approach to the system usage. Usability can be defined in many ways. We see usability broadly according to the ISO 9241 definition: "the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments". We selected 30 people with the aim of carrying out the scenario's tasks in the WS-CAM application. The users collected were students of the Faculty of Computer Science at the Universidad Politécnica de Madrid (UPM). In fact, 20 of the 30 students selected were studying the degree on computer science at the UPM. More specifically, 6 of them were in the first year, 7 of

them were in the fourth year (and therefore they already had a background on distributed systems) and 7 of them were working on their final year project. As for the 10 students remaining, they were MS in computer science, 4 of them were working on a company and 6 of them were PhD students in computer science at the UPM. The scenario selected for this purpose was the described in the previous subsection. In fact, the scenario included only five nodes with the attributes presented in table 1.

Each of these nodes would be added to the environment by using the WS-CAM application. Before starting the experiment, all of them had received the same information as well as a document with a brief description of the set of tasks, rules and schedules to be introduced in the application (figure 5). This experiment was carried out in two steps. As a first step they had to complete the entire scenario in a time (t) ≤ tmax. The second step consisted of responding to a questioner to study the application's usability. His questioner was divided in three parts: The first one was related to the "User's Overall Impression". The second one was related to the "Easiness to Manage the Application". Finally, the last part of the questioner is related to the "User's Interface". Users could select a value from 1 (minimum value) to 5 (maximum value), to response to each and every question.

Table 1. Attributes of the nodes in the experimental scenario

Node	Attributes
First	Memory=3GB;Department=StudyBiological;SO=Windows; Unit=Proteins_Design_Unit
Second	NumberProc=2;Department=SupportBioinformatic; Unit=Proteins_Design_Unit
Third	Memory=3GB;Unit=Biocomputation_Unit;Department=2
Fourth	Center=Nuclear_Center;location=bcn;software=MATLAB; processor=64bits
Fifth	Center=Medical_Investigation;location=bcn;SO=Linux

The results obtained in this experiment were very significant because 90% of the users selected the values from 4 to 5 in the overall impression. More specifically, 60% chose 4 and 30% chose 5 to score their satisfaction, 80% opted for 4 and 10% opted for 5 to count the application efficiency and effectiveness from the user's perspective, and almost 100% decided on 1 to grade the application's complexity, in general, - where 1 represents no complex. As for the second questioner: 90% of the users chose 5 to score the application management in the first time - considering therefore the application very easy to be managed even in the first time-, 70% opted for 4 to count the naturalness of the sequence of steps and almost 100% of the students decided on 5 to grade the application's help to finalise the entire scenario. Finally, 100% of the students opted for 5 to score the suitability of the application's interface, 90% selected 2 to grade possible improvements – where 1 means no improvements and 2 minor improvements, 100% agreed on the terminology used – as they selected 5, and 80% judged the application as pretty intuitive, selected 4 or 5 to respond to the corresponding questions.

In short we can conclude that, in general, it was very easy to run the experiment. None of the participants encountered any problem in completing the scenario. Even although the CAM model was a bit more difficult to be understood by the

Fig. 5. Some Focus and Nimbus rules

undergraduate students, the WS-CAM application was still fairly natural and intuitive, evaluating rather satisfactorily it usability and suitability.

6.3 Performance-Based Validation

These experiments have been carried out in a heterogeneous grid environment made up by a set of virtual organizations with the following computational resources (Table 2). The architecture of these nodes is diverse (from monoprocessor to clusters). In short, there are 12 nodes and 49 processors. In order to generate a set of CPU-bound task in an objective way, the NAS Parallel Benchmark NPB 2.3 has been used [1]. NPB is a suite composed by 7 programs derived from computational fluid dynamics codes. They have been developed at NASA Ames Research Centre in order to measure objectively the performance of highly parallel computers and to compare their performance.

First Experiment: This experiment intends to get a measure of the overhead introduced by the CAM model in the execution of a set of tasks. This scenario describes the ideal conditions for the model: the node N receives a set of consecutive

Table 2. Computational Resources

VO	Node	CPU	Mem.	Disk (GB)	Number of CPUs
Ciemat	gridimadrid	Intel Xeon 2.4GHz	2GB	80	6
University Carlos III	cormoran	Intel Pentium 4 2.40GHz	512 MB	65	1
	Faisan	AMD Duron™ 1350 Mhz	512 MB	46	1
University Complutense	Aquila	AMD Optaron 2400 MHz	1GB	18	2
	Ursa	Intel Pentium 4.3 2.0 GHz	512 MB	60	2
	Cygnus	Intel Pentium 4 3.2.0GHz	2GB	20	1
	Draco	Intel Pentium 4 3.2.0GHz	2GB	20	1
University Politécnica de Madrid	baobab	Intel Xeon 2.40GHz	1GB	20	16
	Brea	Intel Xeon 3.00GHz	1GB	20	16
University Rey Juan Carlos	africa.	IIntel Pentium 4 2.80GHz	1GB	20	1
	Pulsar	Intel Pentium III 1.0 GHz	512MB	30	1
	Artico	Intel Pentium III 450 MHz	128MB	10	1

task execution requests. The N node has full awareness of interaction with the rest of the nodes making up the grid, and therefore this node could consider all those resources to configure the collaborative/cooperative organization according to the corresponding rules. In this case, 100 tasks have been launched with a 3 seconds interval and with 5 processes per task.

The figure 6 presents the overhead introduced by the CAM with regard to the total overhead caused by the management operations of processes. The CAM overhead is related to the calculation of the collaborative/cooperative organization as well as the processes delivery in the environment. The total overhead takes on the previous one plus the consignment of the process to be executed, the monitoring of the process state and the reception of results. As it is possible to appreciate in this figure, the overhead introduced by CAM is almost a constant which doesn't depend on the collaborative/cooperative organization selected. In fact, the percentage of overhead introduced by this model is always lower than the 30% and, in general, is also lower than 20% of the total overhead.

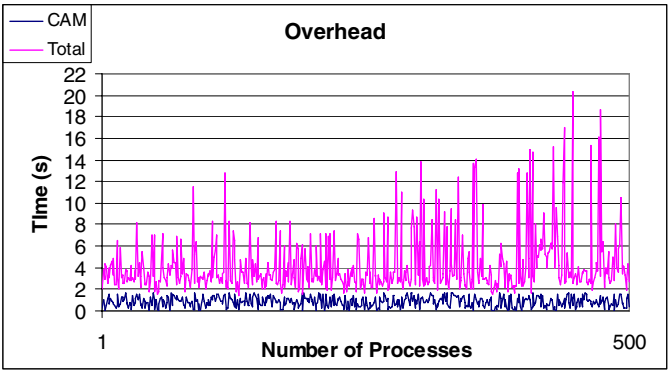

Fig. 6. CAM's overhead with regard to the total overhead for the first experiment

Second Experiment: This scenario raises the non ideal situation in which all the nodes in the grid are been underutilized, and therefore they could receive more processes to be executed, but they are located in different auras. The grid client requests the execution of 100 consecutive tasks, (each one with 5 processes) in the node N. This node has half of the nodes of the grid inside its aura with a distance equal to 1 and the remaining nodes in another aura with a distance equal to 2. The following figures (Figure 7) present the experimental results obtained from this second experiment. In this scenario the aura has been incremented and therefore the model requires a set of messages to carry out the delivery operation. This overhead is reflected in an increment of the communication overhead. This increment gets, in some cases, the 40% of the total overhead associated to the system. However, this overhead never gets the maximum height and therefore the overhead is absolutely delimited for this second scenario.

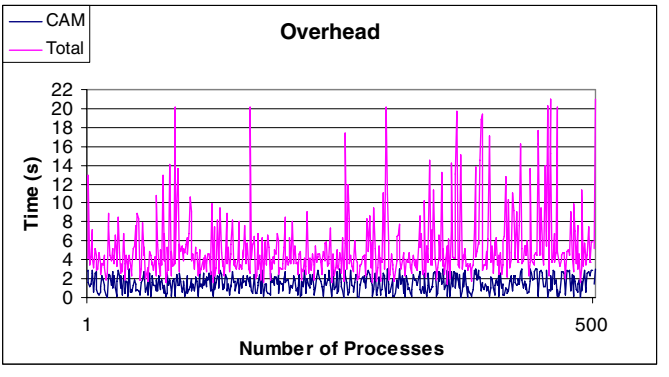

Fig. 7. CAM's overhead with regard to the total overhead for the second experiment

The second metric used to evaluate the model performance is related to the error made by the model in the process of tasks delivery with regard to the optimal tasks distribution, calculated from an a priori knowledge (see Figure 8). As for the tasks delivery, the CAM's model made an excellent work even although the tasks

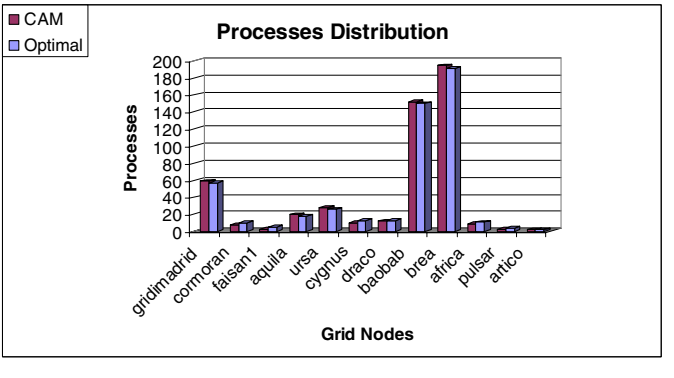

Fig. 8. CAM's tasks delivery vs. the optimal tasks distribution

distribution is dynamic and there is not a priori knowledge. In fact, the maximum relative error is only in 3 processes, an 0,6% over the total.

7 Conclusions and Ongoing Work

This paper presents how manage Virtual Organizations by means of a CAM (Collaborative/Cooperative Awareness Management) model and a business engine for Computer Supported Cooperative Work. The output of this integration is strongly needed: an efficient, flexible and dynamic resources-sharing infrastructure, endorsing interaction by means of a set of rules. CAM manages awareness of interaction by means of a set of rules, optimizing the resources collaboration, promoting the resources cooperation in the environment, and responding to the specific demanded circumstances at a given moment. This paper also describes how CAM works in some specific examples and scenarios, presenting also how the WS-CAM Rules-Based Management Application has been designed, implemented, and validated to tackle Virtual Organizations management in Computer Supported Cooperative Work.

The CAM model described throughout this paper has been evaluated in three different steps: an scenario-based validation to confirm that the model is working properly; an user-based validation to check user preferences, as well as the "real" usefulness of the model; and an performance-based validation to check the efficiency of the functionalities of the model from the more technical perspective, getting satisfactory results in each and every evaluation step.

Acknowledgments

This work has been partially funded by the Government of the Community of Madrid (S-0505/DPI/0235).

References

[1] Bailey, D., Harris, T., Saphir, W., Wijngaart, R., Woo, A., Yarrow, M.: The NAS Parallel Benchmarks 2.0. NASA Technical Report NAS-95-020 (1995)

[2] From Intelligence Community Collaboration, Baseline Study Report (1999), http://collaboration.mitre.org/prail/IC_Collaboration_Baseline_Study_Final_Report/1_0.htm

[3] Batini, C., Mecella, M.: Enabling Italian e-Government Through a Cooperative Architecture. IEEE Computer 34(2) (2001)

[4] Bauer, T., Reichert, M., Dadam, P.: Intra-Subnet Load Balancing In Distributed Workflow Management Systems. Int. Journal of Cooperative Information Systems 12(3), 295–323 (2003)

[5] Benbernou, S., Hacid, M.-S.: Resolution and Constraint Propagation For Semantic Web Services Discovery. Distributed and Parallel Databases 18(1), 65–81 (2005)

[6] Benford, S.D., Fahlén, L.E.: A Spatial Model of Interaction in Large Virtual Environments. In: Proceedings of the Third European Conference on Computer Supported Cooperative Work, Milano, Italy, pp. 109–124. Kluwer Academic Publishers, Dordrecht (1993)

[7] Bernstein, P.A.: Generic Model Management: A Datatase Infrastructure for Schema Manipulation. In: Batini, C., Giunchiglia, F., Giorgini, P., Mecella, M. (eds.) CoopIS 2001. LNCS, vol. 2172, Springer, Heidelberg (2001)
[8] Business Rules Engine (consulted in 2007), http://en.wikipedia.org/wiki/Business_rules
[9] Casati, F., Shan, M.C.: Dynamic and Adaptive Composition of e-Services. Information Systems 6(3) (2001)
[10] Dayal, U., Hsu, M., Ladin, R.: Business Process Coordination: State of the Art, Trends and Open Issues. In: Proc. of the 27th Very Large Databases Conference, Roma, Italy (2001)
[11] Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. In: CSCW. Proceedings of the 4th ACM Conference on Computer-Supported Cooperative Work, Toronto, Ontario, Canada, pp. 107–114 (1992)
[12] Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: Some issues and experiences. Communications of the ACM 34(1) (1991)
[13] Foster I., Kesselman K., Tuecke S.: The Anatomy of the Grid. Editorial: Globus Alliance (consulted September 2004), Web: http://www.globus.org/research/papers/anatomy.pdf
[14] Foster, Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project (2002)
[15] Herrero, P.: Covering Your Back: Intelligent Virtual Agents in Humanitarian Missions Providing Mutual Support. In: CoopIS 2004. LNCS, pp. 391–407. Springer, Heidelberg (2004)
[16] Hwang, G.-H., Lee, Y.-C., Wu, B.-Y.: A Flexible Failure-Recovery Model for Workflow Management Systems. International Journal of Cooperative Information Systems 14(1), 1–24 (2005)
[17] Autonomic Computing (consulted in 2007), http://researchweb.watson.ibm.com/ autonomic/ overview/elements.html
[18] Drools (consulted in 2007), http://markproctor.blogspot.com/2006/05/what-is-rule-engine.html
[19] Kolp, M., Castro, J., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering (to appear in Information Systems) (2002)
[20] Mecella, M., Pernici, B.: Designing Wrapper Components for e-Services in Integrating Heterogeneous Systems. VLDB Journal 10(1) (2001)
[21] Panteli, N., Dibben, M.R.: Revisiting the nature of virtual organisations: reflections on mobile communication systems. Futures 33(5), 379–391 (2001)
[22] Peerbocus, M.S., Tari, Z.: A Workflow-Based Dynamic Scheduling Approach For Web Services Platforms. Iscc 2004, pp. 31–37 (2004)
[23] Wilson, P.: Computer supported cooperative work: an introduction, Intellect books, Oxford (1991)
[24] Yang, J., Heuvel, W.J., Papazoglou, M.P.: Tackling the Challenges of Service Composition in e-Marketplaces. In: Proc. of the 12th Int. Workshop on Research Issues on Data Engineering: Engineering E-Commerce/E-Business Systems, USA (2002)
[25] Greenhalgh, C.: Large Scale Collaborative Virtual Environments, Doctoral Thesis. University of Nottingham (October 1997)