CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2003; **00**:1–12 *Prepared using* cpeauth.cls *[Version: 2002/09/19 v2.02]*

Grids of agents for computer and telecommunication network management

M. D. Assunção[1,*,†], F. L. Koch[2] C. B. Westphall[1]

[1] *Network and Management Laboratory, Federal University of Santa Catarina, Florianópolis, SC, 88049-970, PO 476, Brazil*
[2] *Intelligent Systems Group, University of Utrecht, The Netherlands*

SUMMARY

The centralized system approach for computer and telecommunication network management has been presenting scalability problems along with the growth in the number and diversity of managed equipment. Moreover, the increase in complexity of the services being offered through the networks also contributes for adding extra workload to the management station. The amount of data that must be handled and processed by only one administration point can lead us to a situation where we won't have enough processing and storage power to carry out an efficient job. In this work we present an alternative approach by creating a highly distributed computing environment through the use of grids of autonomous agents to analyze large amounts of data, which reduce the processing costs by optimizing the load distribution and resource utilization. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: grid of agents; grid computing; network management;

1. INTRODUCTION

Grid computing has emerged as an initiative that provides the resource sharing, forming a large scale distributed system and making possible the resolution of complex scientific and commercial problems [1]. By distributing the workload of his applications one can take advantage of a computational and storage capacity that, in a traditional environment, would be financially impracticable.

This resource sharing and aggregation technology differs from others in the Internet on how the component integration takes place. Its aim is to provide a cheap, efficient, easy,

*Correspondence to: Network and Management Laboratory, Federal University of Santa Catarina, Florianópolis, SC, Zip 88049-970, PO 476, Brazil
†E-mail: assuncao@lrg.ufsc.br

Received 1 August 2003
Revised 15 August 2003

pervasive and large-scale access to resources. The basic differences between grid computing and a traditional distributed system are the great heterogeneity of resources, the dynamic environment characteristics and the high latency of the networks that interconnects them. A grid is an aggregation form that can occur in several ways, including through the cooperation and negotiation mechanisms provided by agents.

In the traditional workflow of a network management system we have a situation where data is collected from the network devices through some management protocol; the collected data is analyzed and; transformed into management information. Later on, managers make their analyses and take actions to correct system weaknesses [2]. Rule-based and inference systems could be used to analyze this data, extract necessary information and identify possible problems. However, the consolidation of the collected data into management information is an intensive task and it consumes a great deal of processing power. As the environment grows, the efficiency of a centralized system decreases and the cost of hardware increases proportionally.

We present an alternative architecture, based on grids of agents, distributing the workload of those intensive tasks. We will present some results from our research on grids and its possible use in network management.

This work is divided into 5 sections: In section 2 we present a brief overview on grid computing, agent-based grids and their possible use to assist computer network management; in section 3 we present our proposal for an agent-grid architecture [3]; we compare our architecture with other approaches in section 4; finally, in section 5 we present our conclusions and future works in this line of research.

2. GRID COMPUTING AND GRIDS OF AGENTS

In the computational literature, a grid has appeared as a hardware or software infrastructure capable of grouping components, providing new functionalities, starting from existing components [4].

The grid is basically an infrastructure that helps to integrate resources. The grid of agents can be viewed as an infrastructure or middleware that provides means to software agents to connect to this infrastructure, discover and interact with other agents in a seamless way [5]. In addition a grid of agents can also be composed by a set of agent-based mechanisms like brokers, traders, matchmakers, reasoning that improve traditional systems and provides their integration and collaboration in an easier way.

This area of research has already presented favorable results in scientific works that required intensive computer power not achievable by traditional computing environments. Some shared features of these applications are: the fact they can be decomposed in smaller units; they are not sensitive to communication latency; and their sub-tasks do not require the exchange of a large amount of synchronization messages. We can also find works that provide middleware for the grid computing [6].

In the work [7] an organization is presented based on three levels: computational grid [1],[8], information grid and knowledge grid. When we find the term "grid of agents" we notice that two distinct situations exist: first, we have a grid of agents as a system to connect heterogeneous

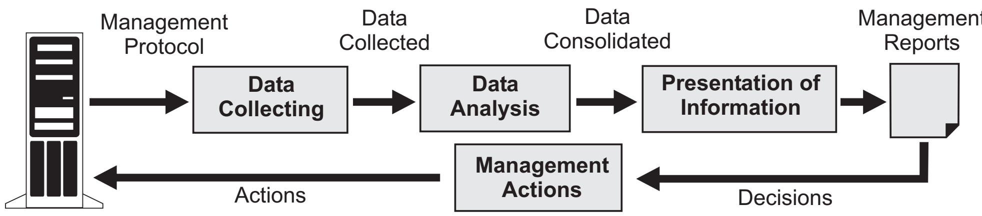

Figure 1. Network management workflow.

components [9] and in the second one we perceive a grid as a kind of middleware that provides interoperability among different agent platforms [10].

It is evident that the concepts involved in agent technologies and multiagent systems can be used for the grid construction. The concepts around agents and multi-agents can be helpful on building grids of agents systems such as cooperation and negotiation concepts helping in the load balance process and local autonomy for the communication latency. In our work we use concepts from previous works to design our own architecture of a platform for grids of agents. Our application is targeted to network management systems. The platform is based on small agents using AgentLight [11] and following FIPA [12] specifications.

2.1. Grids of Agents in Network Management

The complexity of management systems grows proportionally to the size of computer and telecommunication networks. As described in [2], a traditional network management system workflow is presented at Figure 1. In this model, in the first step, the data is collected from managed network devices through a management protocol. After that, the information is analyzed and, finally, condensed, thus creating real management information. Later on, based on this information, the network manager can carry out its tasks and correct weak points in the system. In a network environment formed by a large number of equipments, the volume of collected data that have to be analyzed is proportionally large. The task of transforming this data into information and reports that demonstrate the network operation service conditions or that indicates possible problems could become an intensive task, demanding a great processing power of stations network management.

Many approaches based on Artificial Intelligence concepts have been proposed [13, 14, 2, 15], aiming to increase network management efficiency. Some systems have their functioning based on production rules and inference. Although they assist in the management, these applications also grow in the direction of holding a large number of rules to identify and deal with problems, thus increasing the management system complexity, decreasing its performance and causing a great rise in the hardware cost of management stations.

A negative factor in this scene is the non-existence of parallelism in the analysis. This situation can be improved if we have a large number of agents, possibly located in different stations, analyzing information simultaneously.

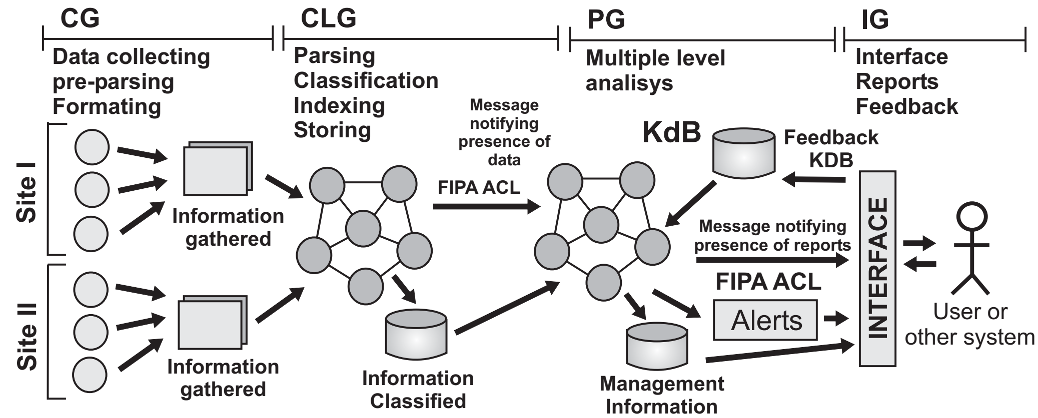

Figure 2. General architecture of the system.

If we think of a network management system as a Grid application with a vast amount of collected data to be compiled into management information we can come up with a scenario where grids of agents could be applied. In this grid, the processing agents can handle huge number of data sets and analyzing rules by distributing the load throughout idle resources that compose the processing grid. The agents can act autonomously and learn new rules when required, thus constantly increasing their 'management knowledge' of the system.

3. AGENT-BASED GRID ARCHITECTURE FOR NETWORK MANAGEMENT

In this section we present our grid of agents architecture to computer and telecommunication network management. Considering the traditional management workflow, we identified some distinct tasks: data collection from network devices; the classification of this data; data analysis to transform it into management information; and information presentation. We will initially present the grids that compose the architecture and then get into design details. Figure 2 shows a sketch of this architecture.

3.1. Collector Agent Grid (CG)

The role of the collector grid of agents is to interface with the network devices, by means of management protocols or other way, collect the data and upload to the next phase of the processing. A collecting agent can have an SNMP (Simple Network Management Protocol) interface or can use a command line utility, for instance.

The information extracted from network devices needs to be normalized into an internal representation format. This representation is made using XML and ontologies [12],[16]. In this case, we guarantee that the grid of agent, which will receive it, could correctly interpret this information.

A collector agent has a knowledge base with rules that allow collecting data using the requested protocol. Those agents can have one or more goals that consist of extracting managed object values from one or more pieces of equipment in the network between time intervals. Just as an agent can interact with one or more devices of the network; many agents can extract data from just one device. Eventually, the collector grid can contain agents that execute some local information analyses. The information collected by the collector grid, already in a standard format, is sent to the classifier grid, through any existing protocol - such as SMTP or HTTP.

3.2. Classifier Agent Grid (CLG)

In a relatively large network, we can have many data collectors. Consequently, we will have data arriving from many types of equipment. The classification grid carries out the task of classifying and storing this information in a more organized and easy-to-retrieve form. A file containing collected data sent by one grid could contain collected values from many managed objects in heterogeneous equipments. It is important to organize this information before carrying through any analysis. From the description above, it is clear that the classifier grid performs parsing, classification, indexing and storing data tasks.

3.3. Processor Agent Grid (PG)

The grid of processing and analysis agents is the heart of the architecture and where the development challenges reside. This component is responsible for processing the collected and stored data into management reports. The main problems that can arise are related to division of analysis activities, resource management, and load balancing and fault tolerance. The result of analysis is management information, formated as management reports and alerts that will be sent to the user by the interface grid.

The processor grid can carry through a series of analyses on multiple levels. Some examples of these analyses, and how they occur are described as follows:

(i) When receiving a message from the classifier grid, indicating the presence of data, the processor grid triggers its rules to look for possible problems. This first analysis does not consider the information previously stored in the database and it does not look for any relationship among the facts.
(ii) The agents belonging to this grid can consolidate data, extracting information previously stored. They can look for problems and group data, so as to generate some information that will be used for assembling management reports.
(iii) On the third analysis level, the agents of the grid can seek relationships among the facts on the database, having a higher-level vision of the information. In this way, they can identify problems that arose through the crossing and not just isolated data.

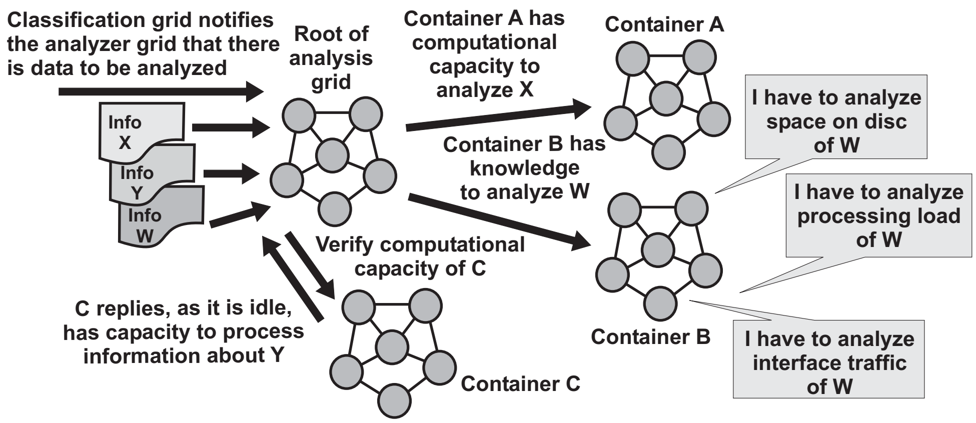

Figure 3. An example of analysis tasks division in the grid.

In this context, the main difficulty is providing a task division in such a way that there is no loss of meaning in the information. This is precisely the role of the classifier grid - it prepares the information so that the analysis tasks can be easily distributed and the processing grid root co-ordinates this distribution, functioning as a broker in the system.

The analysis grid root receives a message from the classifier grid, indicating that there is data to be analyzed and that this analysis needs to be distributed among the containers of the grid. To carry out this distribution, there must be a register that contains the identities of the containers that compose this processing grid and also contains information about the abilities of these containers. In Figure 3 we have an abstract view of the division of analysis activities. As previously mentioned, we can add a new container to the processing grid, with rules to identify some kinds of problems. When the container is added, it announces its services to grid and starts to analyze the information that is assigned to it.

3.4. Interface Agent Grid (IG)

The grid of interface agents is the communication channel between the processor agent grid and the network manager. It is also a way of receiving feedback from the user and supplying it to the system. Through an interface the user can receive the processed information from the processing grid and the alert messages. It is possible to create new reports and to interact with the knowledge base, defining new rules and goals, and to modify existing goals.

This interface must be flexible and multi-protocol, allowing several ways of interaction according to the user's necessity - for example, HTML pages, email, chat, XML/HTTP. It can also work as interface to another systems.

3.5. Load Balancing in Grids of Agents

One advantage of using grids of agents in the collected data analysis is the intelligent distribution of the workload. When receiving a large amount of data, the grid can carry out the distribution between containers, based in the following principles:

(i) Distributing it among those containers that hold the knowledge to process it;
(ii) Utilizing resources that have computational capacity to process it and;
(iii) Using resources that are idle.

To provide this distribution service, first the grid must know the resources that compose the grid at a given moment and their capabilities. This information is stored in the grid root using directory service. When a new container is added, it can register itself and inform the capacity of the resource in which it is running and the services that it is capable of providing. The ontology that is used when the container informs this profile is defined by FIPA [12]. When it is necessary to attribute some activity that involves workload distribution, the root uses this information to select the resources capable of processing it.

When the processing grid receives a message, informing it that there is data to be processed, besides giving this static information, it could request the current profile of the resources or negotiate [17] with containers concerning the possibility of sending information to be processed by them. In this way it can use negotiation protocols established by FIPA [12].

In the same way, the data consolidation and the search for relations between facts can be distributed among several containers in the grid. The analysis grid can select resources capable of performing these tasks and delegate the processing tasks. The root could use the directory information to select capable containers.

4. ARCHITECTURAL ADVANTAGES

We can evidence some advantages of architecture using grids of agents in comparison with a multi-agent architecture like the one presented at Figure 4. In this model we would have collectors extracting data from diverse equipments, a classifier storing this data in a repository and an analyzer agent analyzing this information. Finally, we have an interface with the user, provided by an interface agent.

Each network has a similar structure and there is no relation among different sites. There is no integration in this information; and no high level analysis can be carried out. Moreover there is no kind of workload distribution. The only possible evolution of this system would be the improvement of knowledge bases.

Another disadvantage is that it does not scale well. The only way to increase the power of information analysis is to increase the power of hardware that maintains the system. This is an expensive and unnecessary addition and can add to this the fact that these resources can be allocated and are not being used even when there is no information to be analyzed.

In the architecture we are proposing we would have a more flexible and scalable structure, as it was described in Figure 2.

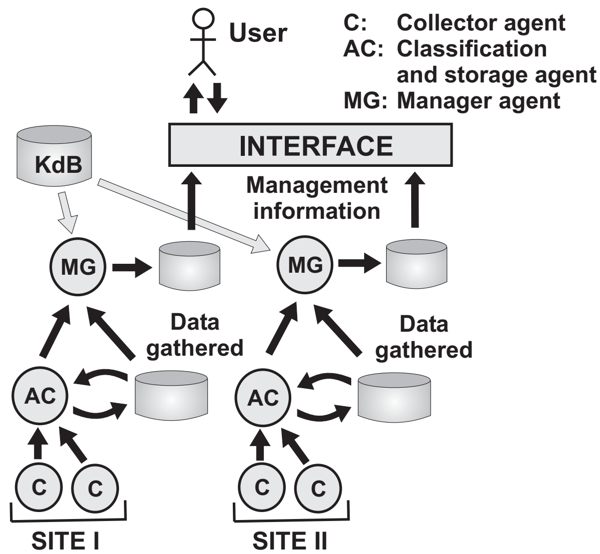

Figure 4. Architecture without grids of agents.

With grid of agents, we merely have a common flow that data must follow. The main advantages are related to the distribution of workload processing and the use of idle network resources. We can also have a number of further advantages:

(i) Extensibility - The architecture can be expanded on any of its levels. New containers or agents can be added; and new goals can be attributed to the existing agents. In the processing grid, we can add new containers to different kinds of equipment, according to the processing and analysis need we wish to meet. A container with several agents can be added to the grid to carry out a more specific type of analysis or to identify a particular type of problem.

(ii) Analysis time reduction - data analysis time can be reduced, because we will have several containers simultaneously analyzing data. In this way a large number of rules and problems can be verified in a shorter period of time. We can affirm this, based on the distribution of the activities and the possibility of aggregating resources in the grid. In a traditional management application, the only way to speed up the processing of this information would be to increase the hardware capacity of the management station.

(iii) A large number of analysis rules - the system can hold a large number of rules. The agents can carry out multiple-level analyses, by using a number of rules to cross check-data, that would not be retained and verified in traditional ways.

(iv) Intelligent load balancing - another advantage of intelligent agents is that they have the knowledge as to how to distribute the processing load, through the discovery of available resources or the utilization of cooperation and negotiation concepts.

(v) A greater processing power and its better distribution - we have a greater processing power resulting from resource aggregation. We also have a more cost-effective use of resources, preventing a situation where some are idle while others are overloaded.
(vi) Shared knowledge - in a system where there is management of several networks, shared knowledge is an important advantage.

In spite of its advantages, we can perceive that the utilization of grids of agents appears to be most attractive when the volume of information to be analyzed on the network is relatively large. In less busy environments, traditional approaches or those based on multi-agent systems, still prove to be more cost-effective.

There is also a point where the utilization of grids of agents is no longer advantageous. As previously mentioned, the division of data and analysis activities may bring about a serious loss of meaning in the information obtained.

4.1. Advantages in Grids of Agents Applications

We can take as an example a network management system whose goal is to monitor the network stations' performance by collecting data such as processor usage, memory availability, available disk space and the list of processes being executed at probing time. Data can be gathered using a management protocol or a command line tool. After collecting the data, a parsing module extracts the relevant information passing it to the analysis module, which runs the inference rules on top of the collected values.

For this example, in order to evaluate the management system performance we will take in consideration three measurable values: communication network with the transmission of management data, processor and disk utilization. For each management activity we attributed a relative value for usage as described in Table I. We divided the requests and inferences uppon the collected data in three different types. After each set of single inferences, a more complex one is carried out. For each host each one of the requests and inferences are carried out in a round. We considered 10 rounds and after the execution of these tasks we reach the results presented on Figure 5 that shows the information about the resources utilization for each management model. These results are based on the possible load balancing capabilities achieved in a grid of agents and are simple calculations obtained through the sum of the relatives resource usage values of the tasks attributed to each host.

In Figure 5a, the model implementing centralized management presents higher network utilization as the data transmitted between the resource and manager station is in raw format, being parsed by the manager itself. Moreover, as there is only one host involved in all activities, its processor becomes the bottleneck.

The second model, whose results are presented at Figure 5b, is the one that implements a multi-agent system. Here, there are more than one data collector hosts, which also carry out parsing tasks where unnecessary information is removed before the data is transmitted to the manager host. Moreover, in this model the collectors have the knowledge about how and when to collect certain value, releasing the manager from making these requests. These features lead to reduction in communication traffic but keep a centralized data analysis structure, which, again, is the system bottleneck.

Table I. Relative times of management tasks

Tasks	CPU	Network	Disc
Request A	10	5	0
Request B	10	10	0
Request C	10	15	0
Parse A	15	0	0
Parse B	15	0	0
Parse C	15	0	0
Storing	10	0	10
Inference A	20	0	50
Inference B	20	0	50
Inference C	20	0	50
Inference AxBxC	40	0	8

Finally, in the model where a grid of agents is applied, besides the distribution of collection and storage tasks, we can also implement a distributed information analysis. For this example, we have 3 collector hosts running the information collection, one host for storage and the two hosts running the inference tasks.

As we can see from Figure 5c, the highly distributed processing implemented through the grid of agents architecture led to extensive work load balancing thus improving resource utilization and allowing higher scalability. Data was obtained taking in account that in the centralized management all the activities are carried out by only a single host, and in this case the values presented are the sum of the times spent with these activities. The network is used to transmit the requests to and responses from the managed devices. The storage and analysis occurs in the same host. In this case, the usage is the sum of the values multiplied by the number of rounds. In the multi-agent scenario, the collection tasks are divided among different hosts and we assume that the parsing activities decrease the size of responses by 20%. This scene presents the storage and analysis being done in an only host and the values of these activities are credited to the manager host, yet. In the grid of agents model, besides the collection tasks, the analysis or inferences are distributed among different hosts, as well as collection tasks. In this scenario the relative values of those activities were assigned to these hosts.

5. CONCLUSIONS AND FUTURE WORKS

In this work we have presented a grid of agents architecture applied to computer network management. We have also shown the concepts involved in grid computing and the advantages of its utilization in the processing and analysis of management data.

Since the concept of grids of agents is relatively new, its utilization in network management must be increasingly more exploited, and this is first work in the area. We consider it important

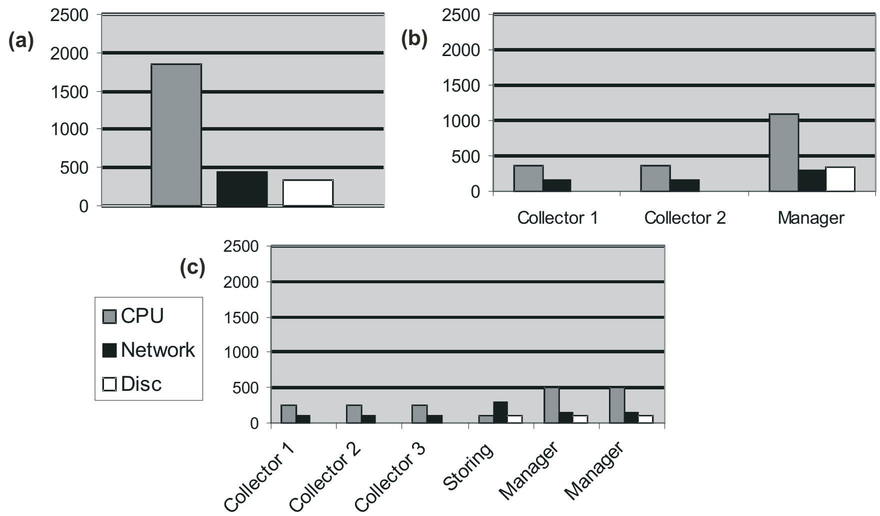

Figure 5. Compared performances from three architectures: (a) centralized management; (b) multiagent with 2 collectors; (c) grid of agents.

to investigate the possibilities of its utilization and propose a grids of agents architecture for network management, validating this approach theoretically and through an implementation of a prototype that is currently being developed. We will enhance the activities carried out up to now, and will proceed with the following work:

(i) Developing better prototypes, showing the advantages of utilizing grids of agents by performing more effective measurements.
(ii) Determining more clearly the point at which the utilization of a grid of agents becomes more advantageous and the point where it ceases to be.
(iii) Conducting studies on load balancing on the processing grid, also seeking efficient means for the division of analysis tasks. To make further measurements of the processing capacity achieved with a processing grid and its advantages compared with traditional techniques.
(iv) Investigating further the utilization of mobile agents in data analysis and in load balancing. Agent mobility allows for a migration of analysis activities attributed to them, improving the utilization of resources.
(v) Improving the efficacy of forms of storage, replication, indexing and recuperation of management data by grids of agents.

We have shown the advantages of applying grids of agents in network management, reducing costs and providing a more efficient processing and analysis through the architecture developed. There is still a great deal to be done and we believe that the continuity and enhancement of this work deserve special attention from the whole scientific community in this area, since it involves a new and promising approach.

ACKNOWLEDGEMENTS

The authors are thankful to Maitreya Natu of the University of Delaware for helping to improve the language and expression of the paper.

REFERENCES

1. Foster I, Kesselman C. *The Grid: Blueprint for a New Computing Infrastructure.* Morgan-Kaufmann, 1999.
2. Koch FL, Westphall CB. Decentralized network management using distributed artificial intelligence. *Journal of Network and Systems Management* 2001; **09**(04):291-313.
3. Assunção M. D, Koch F. L, Westphall C. B. Agent grid architecture applied to computer and telecommunication network management. In *21th Brazilian Symposium on Computer Networks*, Natal, 2003.
4. Rana O. R, Moreau L. Issues in building agent-based computational grids. In *3rd Workshop Of The UK Special Interest Group On Multi-Agent Systems*, Oxford, 2000.
5. Kahn ML, Cicalese CDT. CoABS grid scalability experiments. *Autonomous Agents and Multi-Agent Systems* 2003; **7**(1-2):171-178.
6. Foster I, Kesselman C. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputer Applications* 1997; **11**(02):115-128.
7. Jeffery K. G. Knowledge, information and data, a briefing to the office of science and technology. www.itd.clrc.ac.uk/Publications/1433/KnowledgeInformationData20000124.htm [24 September 2002].
8. Buyya R. Economic-based distributed resource management and scheduling for grid computing. PhD Thesis, School of Computer Science and Software Engineering, Monash University, Melbourne, Australia, 2002.
9. Rana O. R, Walker D. W. The agent grid: Agent based resource integration in problem solving environments. In *16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Lausanne, 2000.
10. Wijngaards NJE, Overeinder BJ, Steen Mv, Brazier FMT. Supporting Internet-scale multi-agent systems, data and knowledge engineering. *Data and Knowledge Engineering* 2002; **41**(2-3):229-245.
11. Koch F. L, Meyer J-J. C. Knowledge based autonomous agents for pervasive computing using agentlight. In *ACM/IFIP/USENIX International Middleware Conference, Middleware 2003 WORK-IN-PROGRESS*, Rio de Janeiro, Brazil, 2003. IEEE Distributed Systems Online. http://dsonline.computer.org/0306/f/koc.htm
12. FIPA - The foundation for intelligent physical agents. http://www.fipa.org [15 March 2002].
13. Rocha M. A, Westphall C. B. Pro-active management of computer networks using artificial intelligence agents and techniques. In *Fifth IFIP/IEEE International Symposium on Integrated Network Management*, San Diego, USA, pages 610-621, 1997.
14. Fuller W. Network management using expert diagnostics. *International Journal of Network Management* 1999; **9**(4):199-208.
15. Vasilakos A. V, Anagnostakis K. G, Pedrycz, W. Application of computational intelligence techniques in active networks. In *Proceedings of the 2001 ACM symposium on Applied computing*, Las Vegas, USA, pages 448-455, 2001.
16. Gruber TR. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* 1995; **43**(5-6):907-928.
17. Shen W, LI Y, Ghenniwa H, Wang C. Adaptive negotiation for agent-based grid computing. In *Proceedings of the AAMAS Workshop on Agentcities: Challenges in Open Agent Environments*, Bologna, Italy, pages 32-36, 2002.