2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

A resource discovery architecture of loosely coupled grid inter-cooperated Virtual Organisations using mobile agents and neural networks

Stelios Sotiriadis, Nik Bessis, Paul Sant, Carsten Maple
Department of Computer Science and Technology
University of Bedfordshire, UK
(stelios.sotiriadis, nik.bessis, paul.sant, carsten.maple) @beds.ac.uk

Abstract – **Grid computing offers a service for sharing resources over uncertain and complex environments. In such multi-participated settings it is essential to make the grid middleware functionality transparent to members by providing the ability to act autonomous and learning from the environment. Parallel to grid, artificial neural networks is a paradigm for processing information; which is inspired by the processes of biological nervous systems. The latter fields can be really promoted from the artificial intelligent agents, which offer an autonomous acting infrastructure of members with proactive and reactive aptitude. As grid is about sharing and allocating resources within uncertain domains, intelligent agents and neural networks may be the mean of achieving an autonomous learning environment of self-motivated members. In this study, we focus on the mobility agents' model aiming to discovery resources dynamically; parallel to the artificial neural networks as a way to achieve the best resource discovery paths. Our work is fundamentally based on the Self-led Critical Friends method; a technique for realizing inter-cooperation among various scales Virtual Organisations (VOs). Their mediator acting nature redirects communication to other parties of different VOs by utilizing a public profile of data stored within VO members.**

Keywords-component: Grid; Self-led Critical Friends; Metadata Snapshot Profile; Mobile Agents; Self Organizing Maps

I. INTRODUCTION

The contribution of this study is that by presenting an association practice among grid, mobility agents and neural networks we aim to fulfil the gaps of the resource discovery and allocation between inter-cooperated grid members. In a typical grid VO it is common that the number of the resources is previously acknowledged and is connected in random topologies. Several internal VO members called Self-led Critical Friends (SCFs) play the role of the mediators to the communication among different VOs as they have interconnections with foreign domain members. It is assumed that security measures and policies have already agreed at the time that SCFs have decided to allow interactions. However, when the environment extends to a large scale, the number of the individuals becomes uncertain as their status change dynamically. Thus, we aim to discuss the prospect of mobile agents as a way to continually discover resources and identify several paths between VO nodes [8]. Eventually, the mobility intelligence of resources will be supportive to the discovery process [6]. By migrating an internal member list to any cooperated VO node the resource discovery will happen periodically and new members may be added to their internal profiles. This is achieved by assigning an address to each member; so we may then collect and update internal member public profile. The storage place of a member is called metadata snapshot profile and is the place of storing addresses, policies, knowledge, physical resources as also communication and execution job times.

In a rational way, node communication is achieved firstly, by attaining policies, followed by pairing knowledge background, and finally, by physical resource and time information coupling [1]. It is essential that by utilizing SCFs we aim to create a neighbourhood of nodes in which communication occurs from one node to another. As previously mentioned mobile agents functionality is to move a part of their self which is a part of a code, to inter-cooperated parties. An open grid requires this strategy as the resource discovery happens in a self-motivated environment. So, it is required to identify ways for achieving the best node selection according to the metadata snapshot profile information. A neural network sees the grid nodes as artificial neurons, with simple elements processing and a high degree of interconnections. Their proactive nature makes them ideal for such complex infrastructures by providing the ability to learn from the environment. More specifically, they learn from experiences, because any change to the connection weights among two VO members, causes the network to learn the solution to a problem. By assigning a weight to each pairwise path we aim to identify a weighted route among inter-collaborated members. We suggest that communication times among each loosely coupled interaction are stored within the public profile as the weight of the path. The strength of connection between the nodes is a weight-value for the specific connection.

Finally, the entire environment learns the new knowledge by adjusting these connection weights. In the following sections we present the motivation of the study

978-0-7695-4237-9/10 $26.00 © 2010 IEEE
DOI 10.1109/3PGCIC.2010.53

(Section 2) and the related works (Section 3). After, we present the architecture of the specific model (Section 4), the mobility agents' functionality (Section 5), the Self Organizing Maps architecture and their operation milestones aiming to the inter-collaboration infrastructure (Section 6, 7). Finally we conclude our work by analyzing the future challenges of the proposed method (Section 8).

II. MOTIVATION

Agents and grid serve a very important role in the context of communication among heterogeneous and geographically distributed resources. In this direction the method of SCFs based on the open grid idea, introduced by [4] defines a novel model of inter-collaborated VOs of loosely coupled multi-institutional VOs. By considering the grid as solving problems in dynamic and uncertain distributed domains, mobile agents may be the mean to address problems by acting flexibly in such environments. In [3] we have presented a theoretical approach of different decentralized agent models. The approaches may be utilized depending on the scale of VOs in order to achieve continuous resource discovery. By using their SCF ability as intermediate stations in the communication extend their acting environment. In essence, each SCF plays the role of a mediator by reflecting the information to different nodes as much as possible in order to maximize the grid members.

The algorithm of neural networks can be adapted in the aforementioned situation in which designers are seeking for distribution, autonomy, heterogeneity, dynamic resources. More specifically, neural networks may supply a resource selection scheme of dealing powerfully the non linear problem of the best resource choice. By training the input set the algorithm starts with small random weights and finally finds the best solution by changing the connection weights. There are two types of learning in neural networks; the supervised and unsupervised learning. In the first, each node has a specific way of organizing themselves according to a training set given by the member. The second method does not provide a sample ideal output to the neurons so it is required to choose a predictive strategy in order to select the best paths. In our work we aim to provide a transparent setting so the hidden neurons must find a way to organize the network by themselves without any assist from the member. We select the Self Organizing Maps (SOMs), also known as Kohonen network paradigm, of competitive learning by allowing more than one unit to update its weights on a given pattern.

III. RELATED WORKS

A number of authors have described the applicability of agents in the grid context [7, 8, 9, 10, and 11]. They suggest that a successful interoperable infrastructure requires agents as they form a basic component for achieving decentralization. Various resource discovery approaches, including agents have been introduced during the last years, however [6] suggest that the methods were mainly based on the Query and Agent based technique. In the first approach, the resource information is queried for availability, and in the second approach agents periodically monitor information as also communicate with other agents for discovering resources. The major difference between the aforementioned methods is that an Agent is acting on its own decisions by using its internal logic in conjunction with the Query based approach which resides within a fixed query engine.

Related works also include the discussion of the agents' possibilities to act as web services in the concept of grid [13], as also a semantically integrated model of autonomous agents as discussed in [14]. Both models provide a correct level of autonomy based on the multi-agents paradigm; however, when the system extends to a large scale the models do not provide an accurate heterogeneous architecture. In this direction [12, 16] suggest that at the present, most agent based systems are immature and few standards have been developed such as the Foundation for Intelligent Agents Framework (FIPA). FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies [15].

Internal information of members is stored to a public profile of members which is available to any VO participant upon request and is called the metadata snapshot profile [5]. Agents are capable of collecting internal data and processing them in order to find several routes for approaching interconnected members. The resource selection is considered using the Self Organisation Map network which utilizes metadata snapshot profile information about communication times stored from a history list of previous interactions. In other words, it is a feedback of a specific route awarded by the agent functionality. Relevant feedback processes have been effectively used in [17] which refer to a centralized solution and suggest that feedback process may be better adapted than the more general data collection usage.

In [18] Self Organisation neural network architecture is presented for intentional planning agents. The authors discuss that some of the challenges in using neural networks are the basis for agents and explain how beliefs and intentions of agents can be used with a self-organising neural network. Finally, they suggest that more in depth studies are required to obtain a more effectively solution of network planning parameters. In previous works we have discussed the applicability of genetic algorithms in the area of best path exploration among loosely coupled connections of VO nodes [2]. In the same direction, in this study we aim to utilize the fact that neural networks may offer an autonomous acting environment of members by utilizing information stored within each member public profile.

IV. THE RESOURCE DISCOVERY MODEL

The architecture derives from the combination of internal VOs policies and knowledge. We first aim to achieve an authorization procedure among inter-collaborated VOs as also define the knowledge coupling level. Then resource discovery happens by utilizing mobile agents capable of performing migration to any interconnected member. Finally, using a self organisation map algorithm we are seeking for the best node selection according to information stored within each individual public profile. Figure 1 illustrates the aforementioned standards.

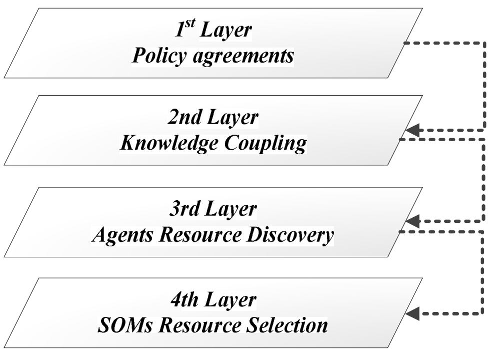

Figure 1: Resource Discovery Architecture

Within each VO member an agent is created and waits for communication from other members. More specifically, agent specification includes the following:

1st Layer: Policy Management: The agent collects internal data concerning policies and securities. It is the process in which a member defines its accessibility factor. Three types of agreements are expected to be utilized after an inter-collaboration occurs. In [1] we have classified agreements as clique contracts, SCF contracts, and reference to the SCFs contracts, which are stored within a node's public profile. First, the clique contracts are agreements signed from the internal VO nodes and their functionality is to arrange interactions for newly added nodes to the clique. Secondly, two nodes that acts as SCFs and belonging to different VOs, agree for SCFs contracts as indication for inter-collaboration of both parties. Finally, the reference to the SCFs contracts are a way of achieving interaction with foreign nodes of different VOs using the SCFs as mediators. In other words, the reference contracts are suggestions to a SCF contract which is settled by the clique member in case a SCF needs to communicate with a node which is not aware of its existence. Within such contract interactions agents of parties interchange their contract agreements in order to achieve the interoperable environment.

2nd Layer: The public snapshot profile of a node should be able to provide information concerning its knowledge and capabilities. The pairing procedure is important as the node should be able to decide which node matches the needed job delegation before the resource discovery and selection process. The agent functionality here is to couple the job description. In addition, the internal agent waits for communication from any interconnected member. If a member requests for a job the agent checks the job and decides the fact that current node will became a candidate node for the specific job.

3rd Layer: Agents resource discovery happens within a VO in a continually base. This course proceeds by travelling through a domain, starting from a random node, and after visiting each member terminates at the same node. It is essential that VO members contain an agent waiting for connection as the previous layers. At a random time an agent starts migration from a specific node and starts a journey to any interconnected node, including the SCFs. By currying internal data of the public profile it travels through any domain and updates an address list as well as compares policy agreements and knowledge coupling. During the route, each visited member public profile is updated and discovery occurs dynamically. The functionality is achieved in this multi-agents cooperation network be agents teamwork with other agents within various VOs. Figure 2 illustrates the procedure.

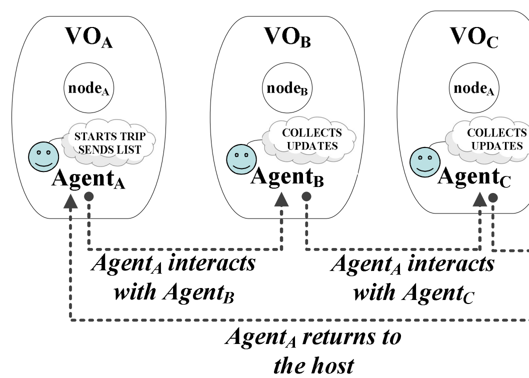

Figure 2: Agents resource discovery

4th Layer: The Self Organisation Map is selected as an optimization technique for selecting the best paths for the continually resource discovery. Each time a journey occurs the agents are travelling in pair-wise distances, so they collect the distance weight from node to node. Having in mind the loosely coupled interactions among different VO parties the paths may be changed dynamically so each time that a journey finishes a new Self Organisation Map is generated for a best path decision. By using an unsupervised competitive learning we may provide a way of representing multidimensional data in much lower dimensional spaces. One of the most interesting aspects of SOMs is that they learn to classify data without any supervision. Into the following sections we discuss the mobile agents paradigm; by focusing on the migration phase of agents. Then we employ the SOM method for deciding the best paths for achieving best routes selection in order to minimize the interval time of the journey.

V. THE MIGRATION AGENT RESOURCE DISCOVER MODEL

The resource discovery method is based on the Java Agent Development Framework (JADE). Each member of a VO is assumed that includes the specific middleware which offers platforms for creating internal and travelling agents. There are two types of platforms a) the intra-platform and b) inter-platform mobility [15]. The intra-platform allows agents to be utilized within an already known environment, in contrast with the inter-platform in which agents may communicate with any platform. In this study, we suggest that inter-platform mobility may offer significant advantages in the area of grid, and plays an important role in the development of an open grid standard. This includes asynchronous processing of data and autonomy to VO members. In our view, the resource discovery in an uncertain environment includes the following agents:

A. Internal member agent

It is the mean of creating the agent service as a part of code which waits for a travelling agent connection. Furthermore, the middleware is based on JADE framework and it is installed to each grid resource, creates an agent that waits to be stimulated by a travelling agents. At the current stage the functionality includes an enquiry of the travelling agent about new members; comparing the already stored addresses of the travelling agent. It is a manner of security in which the JADE framework does not allow agents to directly have access to internal member resources. For that reason, the internal members' agent collects information about physical resources and sends the information back to the travelling agent. Figure 3 demonstrates the interaction of an internal and travelling agent at the time of stimulation.

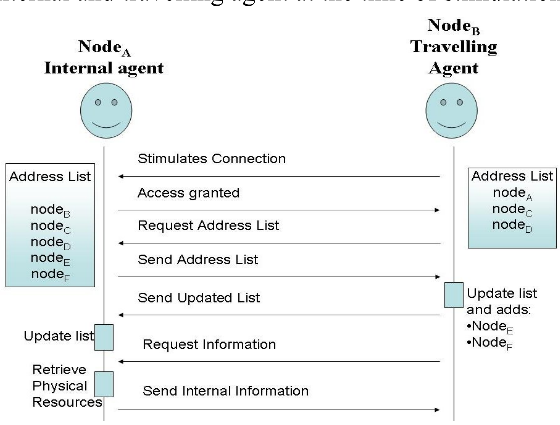

Figure 3: Agents stimulation

B. Travelling agent

As travelling agents, we denote the part of the code that can be easily moved to any interconnected member. The idea is simple; the agent completes a route to any interconnected member in a mode of visiting each individual only one time. The agent stimulates the waiting internal agents and exchanges information. Moreover, the travelling agent contains a file including the addresses of members, as also the physical resources extracted from the internal members.

Each time a visit occurs, the list is updated with new knowledge about the domain as also physical resource curried among the domain parties. In our study the resource discovery happens directly by the autonomous agents and new addresses are stored in the list in a serialized way. In other words, if the travelling agent finds a new node then visits the new address at the end of the route. It is essential that each member can create internal and travelling agents as also multi-agents creation can happen within several VOs.

After a successful route the travelling agent returns back to the staring point. The infrastructure provides a fault tolerance acting environment in which agents can address and aid with fault conditions by moving to an alternative platform when problems are detected. So, it is almost certain that the performance of the travelling agents will not be affected by any circumstances. The interval time of a route is described as the total period that an agent requires for completing a journey through any connected individual. According to [15] agent itineraries can be distinguished in two basic types, a) static itineraries which are determined at agent creation time and b) dynamic itineraries which are determined during the execution time and forming according to agent needs and desires. In our study, we employ the dynamic itineraries as the inter-platform environment of the inter-collaborated VOs may change at any time. In the case of a centralized VO the static routes may be selected as they don't change over time, so an interval time of route may be defined by the central manager. On the other hand in decentralized grid the autonomous agent's environment indicates that migration among several platforms is subject to the accessibility factor of members. So, in our case the interval time of a route may be calculated, however, it is expected to change as the travelling agent discovers new members. Finally, the inter-cooperation model treats interoperable members of different VOs as the same, by respecting the policies articulated by members. In section 4 we have discussed the policies that may be spread across the network by using agents; however, in this study we assume that a good willing environment of VOs is selected aiming to an open grid infrastructure.

Physical resources of members are comprised from the hardware potentials of a member. These include CPU power, the physical memory, the available physical memory, the number of hard disks and their storage availability as also the operating system environment. Moreover, after agent stimulation, the travelling agent collects the ping time of the communication among two members as their pairwise distances. Based on that variable we may provide a self organisation map of best routes within a VO. So a new functionality of the travelling agent will be to stimulate an internal agent procedure in order to ping to any interconnected member. Finally, ping data from member profile will be carried out to the starting point.

VI. THE SELF ORGANISATION MAP ALGORITHM

The selected neural network is introduced by Kohonen 1984 [19] and provides the learning ability of a neural network without being supervised. In this study a SOM learns to classify the training data without any external supervision. As discussed in previous parts node distances have been calculated utilizing agents. The distance is measured by using the ping variable. In general, Kohonen networks consists of two layers; the first is the input layer and the second is m*m dimensional lattice of nodes. Each node has a specific topological position and it is a vector of x,y coordinates in the lattice. So a vector consists of a training set **S**, which are the nodes and each one with a corresponding weight value **W**. So the input set is comprised as in (1).

$$S = (s_1, s_2, s_3, ..., s_n) \quad (1)$$

The weights which are connected to each lattice node **i** in the input layer are as in (2).

$$W = (w_1, w_2, w_3, ...w_{in}) \quad (2)$$

The SOM does not require any training, instead when the weights are matching with the input layer that area of the lattice is selectively optimized to more closely resemble the data for the class that input vector is a member. When a value is assigned at the input layer neurons get the appropriate value. Neurons in second layer sum up the input values from the first layer. Finally, a neuron is selected randomly with the value 1 and all other neurons have value 0. Next, the neural network is trained and the weights are changed until their values are the same as the input set. As in other networks initialisation values are randomly chosen between 0 and 1. The training occurs as following:

1. The node's weights are initialized and a vector of input values is selected randomly from the set of training data. Randomly chosen values are presented to the lattice.
2. Nodes' calculates their weight and the one with the most similarities to the input vector is selected. This node is called the BMU and it is the winning node.
3. The algorithm calculates the radius of the affected neighbourhood. Typically the value starts large and diminishes at each removal step. Any nodes found within this radius are deemed to be inside the BMU's neighbourhood.
4. The weights of nodes found in step 3 are adjusted to make them like the input vector values. It is essential that nodes closer to the wining node alter their value significantly.
5. Finally we repeat step **1** for **n** iterations.

In order to calculate the best matching unit, we select the method of iterating though all nodes and calculate the Euclidean distance between each node's weight vector and the current input vector. In n dimensions, the Euclidean distance between two points x and y is calculated by the equation (3).

$$\text{Distance} = \sqrt{(\textstyle\sum_i=1^n (x_i-y_i)^2)} \quad (3)$$

where x_i (or y_i) is the coordinate of x (or y) in dimension i. The winning node is the one closest to the value of the input vector. So the selected weight is given by the equation (4).

$$\text{BMU} = \min\{\text{Distance}\} \quad (4)$$

After the best neuron selection we define the neighbourhood around the BMU. As neighbourhood we define the lattice nodes which are closer to the winning. In the start of the procedure the neighbourhood of the wining node C is the Nc and it is represented by a big square. However, while the weights are adjusted the neighbourhood is minimized and finally it has the unique value of C. The equation for adjusting the weights for i belongs to the neighbourhood is given in (5).

$$\Delta w_i = a(s - w_i) \quad (5)$$

In any other case the weight altering is 0. As $\boldsymbol{\alpha}$ we denote the learning ability measure. The equation for minimizing the value is given by the equation (6).

$$\alpha_t = \alpha_0 \, (1\text{-}1/T) \quad (6)$$

where $\boldsymbol{t}$ is the number of iteration, $\boldsymbol{T}$ the total number of iterations, and $\boldsymbol{\alpha_0}$ the initial value. So, to conclude the network learns by repeatedly presenting a pattern at the input layer. Then by selecting the winning node adjusts each of the weights which are closer to the input vector. Finally, the after each learning iteration, the neighbourhood size is automatically shrunk by a small amount.

VII. USING SOM TO SELECT THE BEST PATH

In an inter-collaborated environment the number of nodes that agents have to visit is unknown at every time of a travelling agent initialization. In such environments it is essential to find the best travelling in order to minimize the interval time of the complete route. In order to achieve it we suggest that the Kohonen network should calculate routes for the already discovered members. More specifically the list of visited members will be generated at the host member as a vector of addresses. Newly added members will be added to the end of the list and will be visited lastly. The duration weight among each node will be selected as the ping time among members. As stated in previously sections each internal agent contains the ping times to any interconnected member, so any possible route will be considered based on that fact. The idea of the travelling salesman problem fits to our problem as the number of visits is comprised by a set of n cities, or n nodes in our study. The salesman needs to visit each city only once and finally returns to the starting point. The problem has been characterized as an NP-hard problem because of the huge search space of n cities. In our study we aim to optimize the agent path among n members in order to reduce the interval time as much as possible. The distance among them is the ping time that has previously discovered by the travelling agent model. Then, by applying the SOM algorithm we can discover the best node path with the minimum distance. The SOM algorithm for selecting the best resource discovery path is the following:

```
Start the training-SOM
Begin
        randomize all neurons weights
        for (i=1 to t) do
        start
                get a random input pattern
                locate the BMU
                locate the BMU neighbourhood
                adjust BMU neighbohood nodes weights
                minimize the a and N_c
        end
end
```

The above SOM algorithm returns the best path for a continually discovery of newly added resources to each interconnected VO.

VIII. CONCLUSION

In this study we have discussed the association of mobility agents and self organisation maps in order to assist the resource discovery of inter-collaborated nodes. First the mobility agents with their proactive nature may travel across unknown domains and by exchanging information with internal agents updates the address list of members. The routes for resource discovery are dynamically updated; however a solution in order to minimize the newly generated routes is a Kohonen network that utilized the internal public profile information of members. The future work includes an analytic discussion of the mobility and SOM paradigm. Moreover, it is essential to approach the representation of the metadata snapshot profile in a semantically way.

REFERENCES

[1] Sotiriadis, S., Bessis, N., Huang, Y., Sant, P., Maple, C., "Defining minimum requirements of inter-collaborated nodes by measuring the heaviness of node interactions". in: International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010), IEEE, Krakow, Poland, February 2010.

[2] Sotiriadis, S., Bessis, N., Sant, P., Maple, C., "Encoding minimum requirements of ad hoc inter-connected grid virtual organisations using a genetic algorithm infrastructure". in: IADIS multi confernce on computer science and information Systems (MCCSIS 2010), Freiburg, Germany, July 2010. {To appear}

[3] Sotiriadis, S., Bessis, N., Sant, P., Maple, C., "Towards decentralized grid agent models for continuous resource discovery of interoperable grid Virtual Organisations". in: The third international conference on the application of digital information and web technologies (ICADIWT 2010), Istanbul, Turkey, July 2010. {To appear}

[4] Huang, Y., Bessis, N., Brocco, A., Sotiriadis, S., Courant, M., Kuonen, P., Hisbrunner, B., "Towards an integrated vision across inter-cooperative grid virtual organizations". in: Future Generation Information Technology (FGIT 2009), pp.120-128, Springer LNCS, Jeju island, Korea, 2009.

[5] Huang, Y., Bessis, N., Kuonen, P., Brocco, A., Courant, M., Hirsbrunner, B., "Using Metadata Snapshots for Extending Ant-based Resource Discovery Functionality in Inter-cooperative grid Communities". In: International Conference on Evolving Internet (INTERNET 2009), IEEE, Cannes/La Bocca, France, August 2009.

[6] Athanaileas, T. E., Tselikas, N. D., Tsoulos, G. V., Kaklamani, D. I., "An agent-based framework for integrating mobility into grid services", Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, Innsburck, Austria, 2008

[7] Iosup, A., Epema, H. J. D., Tannnebaum, T., Farrellee, M., Linvy, M., "Inter-Operating Grids through MatchMaking", in: 2007 ACM/IEEE SC07 Conference, pp. 1-12, November 2007

[8] Foster, I., Jennings, N. R. and Kesselman, C. (2004) Brain meets brawn: why Grid and agents need each other. In: 3rd International Conference on Autonomous Agents and Multi-Agent Systems, 2004, New York, USA. pp. 8-15

[9] Cox, P. D., Al-Nashif, Y., Hariri, S., "Application of Autonomic Agents for Global Information Grid Management and Security", in: Summer Computer Simulation Conference 2007 (SCSC 2007), San Diego, USA, 2007

[10] Zerfiridis, K.G. and Karatza, H.D., "Mobile Agents as a Middleware for Data Dissemination". Neural, Parallel & Scientific Computations, Dynamic Publishers, Atlanta, 2002, Vol. 10 (3), pp. 313-323.

[11] Zerfiridis, K.G. and Karatza, H.D., "Brute Force Web Search for Wireless Devices Using Mobile Agents", in: Journal of Systems and Software, Elsevier, 69(1-2) pp. 195-206, 2004.

[12] Sharma, A. and Bawa, S., " Comparative Analysis of Resource Discovery Approaches in Grid Computing", in: Journal of Computers, Vol 3(5), pp. 60-64, 2008.

[13] Zehreddine, W. and Mahmoud, Q. H., "An agent based approach to composite mobile web services", 19th International Conference on Advanced Information Networking and Applications, 2005. AINA 2005, vol 2, March 2005, pp. 189-192

[14] Bellavista, P., Corradi, A., Monti, S., "Integrating Web Services and mobile agent systems.",25th IEEE International Conference on Distributed Computing Systems Workshops, 2005, June 2005, pp. 283-290

[15] The Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org

[16] Bellifemine, F., Caire, G., Poggi, A., Rimassa, G., "JADE: A software framework for developing multi-agent applications", Lecture Notes in Computer Science, Intelligent Agents VII Agent Theories Architectures and Languages, pp. 42-47. Springer Berlin / Heidelberg (2001)

[17] Bellifemine, F. and Rimassa, G. 2001. Developing multi-agent systems with a FIPA-compliant agent framework. Softw. Pract. Exper. 31, 2 (Feb. 2001)

[18] Subagdja, B. and Tan, A. 2009. A self-organizing neural network architecture for intentional planning agents. In Proceedings of the 8th international Conference on Autonomous Agents and Multiagent Systems - Volume 2 (Budapest, Hungary, May 10 - 15, 2009). International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1081-1088

[19] Kohonen, T., Self-Organizing Maps, Springer, New York, 3rd edition, 2000