Advanced Science and Technology Letters
Vol.49 (SoftTech 2014), pp.65-72
http://dx.doi.org/10.14257/astl.2014.49.15

The Replanification Model of the agent-platform PANGEA

Carolina Zato[1], Sara Rodríguez[1], Javier Bajo[1] and Juan M. Corchado[1]

[1]Department of Computer Science and Automation
Science Faculty, University of Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{carol_zato; srg; corchado}@usal.es
[2]Department of Artificial Intelligence, Computer Science Faculty
Technical University of Madrid
Campus Montegancedo, Boadilla del Monte, 28660, Madrid, Spain
jbajo@fi.upm.es

Abstract. New trends in multi-agent systems call for self-adaptation and high dynamics, hence the new model of open MAS or virtual organization of agents. This paper describes a system based on virtual organizations of agents that allows planning of tasks that minimizes the resources necessary for its accomplishment and obtains the maximum profits. For this purpose, a genetic algorithm , queuing theory, and CBR are used to obtain an efficient distribution. This mechanism is integrated in the PANGEA platform, an agent platform to develop open multi-agent systems, specifically those including organizational aspects such as virtual agent organizations. The platform allows the integral management of organizations and offers tools to the end user.

Keywords: open MAS, virtual organization of agents, planification, task allocation, PANGEA platform.

1 Introduction

A multi-agent system (MAS) is basically a network of organizations focused on solving problems and working together to find answers to problems that are beyond the individual capabilities or knowledge of each entity [3]. In our case, these entities are CBR-BDI agents, which gets its name from the BDI architecture with a CBR reasoning system [7][8]. The open MAS [4] should allow the participation of heterogeneous agents, which change over time, with architectures and even with different languages. For this reason, we cannot rely on agents´ behaviour, when it is necessary to establish controls on the basis of norms or social rules. For this and because of the characteristics of open environments, new approaches are needed to support the evolution of systems, and to facilitate their growth and run-time updates especially due to the dynamics of open environments. This is one of the reasons that encourage the use of virtual organizations (VO). A VO [5] is an open system designed for grouping, for the collaboration of heterogeneous entities and where there is a

ISSN: 2287-1233 ASTL

separation between form and function that defines their behavior. The concept of organization is seen as a promising solution to manage the coordination of the agents and control their behaviors and actions. Every organization needs coordination support to determine explicitly how to organize and carry out the actions and tasks within it. In this work, thanks to the integration of this method in PANGEA [14], organizations are comprised of organizational units of an agent of hierarchy type where the supervisor has control over other members, coordinating the tasks, centralizing planning and decision-making.

Planning and distribution of tasks is a problem that can be encountered in various activities [1][2]. These works seek to optimize the allocation of resources to maximize efficiency and minimize costs. The main problem arises from the dynamism of work scenarios and the difficulty of finding a balance between time spent on planning and on implementing plans which particularly in these environments is the key to adapt to the needs of the organization where you have to replan frequently. Therefore, it is necessary to create a system that predicts demand for resources so it can perform tasks allocation maximizing benefits and minimizing delays.

This paper proposes a system able to estimate work demands in order to estimate the number of resources needed and according to these claims carry out a work-resources distribution to maximize benefits and minimize delays. The planning model is used by the PANGEA platform to distribute the tasks among the available agents.

2 Proposed Model

The model proposed in this paper focuses on developing a planning mechanism to coordinate the agents found in the VOs. Thus, first we will set out the roles that these agents can take:

- **Processor role.** Responsible for carrying out the activities required for each specific task. For this reason, the responsible agent will specialize depending on the type of tasks the system must solve.
- **Planner role.** Design the overall plan to be implemented by the organization. Sets the number of processor agents and makes the distribution of tasks depending on the role they play. Replan depending on the size of the input queue or inability to comply with a plan.
- **Distributor role:** Distributing tasks according to its completion by the agents and checks that each task is being processed within time limits to serve the plan.
- **Manager role:** This agent manages all the information of the task and communicates to the user.

2.1 Dynamic Planning Roles

The number of agents that should be available in the system is estimated dynamically. It is intended that the number of agents suits demand to ensure that the system utilization factor is less than 1. This estimate will be done through the use of queuing theory in a model M / G / s, where the arrival rate follows a Poisson distribution (the most commonly used in similar work [10]), the exponential service and the existence of multiple servers (agents).

The average waiting time and the overall average length is reduced to calculating the average values calculated for each of the tasks. In the case of the M / G / s model where s = 1,2,3, ... is the number of agents and given an arrival rate = = cte, the service rate when there are n processes is defined by the following equation (2) [9].

(2)

Where represents the average service rate for s available agents. This value depends on both the agents and the machine found. Assuming that the system is in a stable condition, ie it meets the utilization factor , the probability that n tasks exists (Pn) in the system is given by equation (4) [9].

P $n = 0, ?1,?.., s - 1$ (4)

$s!$

Where: (5)

$= \frac{1}{1}$

$n\,p$

$n!\,p \quad sp$

s

p

Having defined the probability that n tasks exist in the system, you can define the number of tasks in the system queue and average waiting time of tasks in the tail of the system (6) [9].

$s \quad) \quad L_P \quad å_n \quad = \quad {}_q \quad =(\quad s^n \quad \frac{P}{m} \quad θ ø$

sm

To determine the optimal number of agents we make an estimate that minimizes the cost function that depends on the number of agents used and on the waiting time in the queue. The function is defined in a particular way for each service depending on the actual costs of each agent in the system, though the following benefit function is provided (7).

(7)

(6)

$L_q \qquad \frac{l}{} \; \frac{1}{}$

$ø \; ÷$

$q \; □ \qquad ç$

(8)

Where k is a constant associated with the cost of having an agent working, the benefit of performing the task, is the average time to complete the task obtained from the service rate, the average time to execute a task. If we overpass the conditions of stability, f_b is counted only up to the utilization factor 1. The utilization factor varies according to the new services added to the queue till it reaches the utilization factor of 1.

Following the cost function given in (7), we introduce the global cost function (9) that takes into account the implementation of the various services.

min ? (

(9)

2.2 Task Assigment

Once the number of starting agents is considered to minimize costs, we proceed to make an allocation of tasks between the available agents. If the system utilization factor does not exceed the value of 1, the distribution of tasks among agents is performed so as to ensure as far as possible that it can perform assigned tasks in case of delays or the time to perform a task increases. It is performed so as to maximize the following function (10):

k

(10)

k

k

Where with the maximum time for completion of the task i, the cumulative time to perform the tasks i-1 above and finally the time to run the task i, which is customized according to the agent selected and calculated from the average value of previously executed tasks. Minimizing the differences get all the tasks to have a uniform distribution of the remaining time so it gets easier to achieve them.

If the system utilization factor is greater than 1, the aptitude function is redefined to minimize possible losses of the work already done

(12

) As in the previous case to know the value of each B_i it is necessary to establish the order of execution of the procedures. If at the time of completing this task the value of B_i was not taken into account.

The chromosome encoding is performed so that each gene is composed of the elements listed by t_i identified in (1). The crossover operator is defined similarly to the multi-junction used in other problems such as TSP. Mutation operators define various modes that will be executed randomly, and just those mutations that improve the aptitude of chromosomes will be selected. Definite mutation operators are to: exchange order of tasks, exchange of assigning contiguous tasks and changing the allocation of a task. Elitism operator is defined to keep the percentage of efficient solutions in every generation of population and population size as a constant this involves the replacement of parents by the children chromosomes in generations with the exception to remain with elite chromosomes. The roulette selection is the criteria chosen for this.

3 PANGEA Integration

As previously mentioned, a platform that can integrally create, manage and control VOs was developed for this study. In general terms, the proposed platform includes the following characteristics:

- Different models of agents, including a BDI and CBR-BDI architecture [12].
- Ability to control the life cycle of agents with graphic tools.
- A communication protocol that allows broadcast communication, multicast according to the roles or suborganizations, or agent to agent.
- A debugging tool.
- Module for interacting with FIPA-ACL agents.
- Service management and tools for discovering services.
- Web services.
- Flexibility in allowing organizations with any topology and suborganizations.
- Organization management.
- Services for dynamically reorganizing the organization [13].
- Services for distributing tasks and balancing the workload [13].
- A business rules engine to ensure compliance with the standards established for the proper operation of the organization.
- Java programming and easily extensible.
- Possibility of having agents in various platforms (Windows, Linux, MacOS, Android and IOS)
- Interface to oversee the organizations.

When launching the main container of execution, the communication system is initiated; the agent platform then automatically provides the agents shown in Figure 1 to facilitate the control of the organization:

- **OrganizationManager**: the agent responsible for the actual management of organizations and suborganizations. It is responsible for verifying the entry and exit of agents, and for assigning roles. To carry out these tasks, it works with the OrganizationAgent, which is a specialized version of this agent.

- **OrganizationAgent**: it is a specialized version of the OrganizationManager, which is introduced automatically in each suborganization to help the OrganizationManager and avoid its overload.
- **InformationAgent**: the agent responsible for accessing the database containing all pertinent system information.
- **ServiceAgent**: the agent responsible for recording and controlling the operation of services offered by the agents. It works as the Directory Facilitator defined in the FIPA standar.
- **NormAgent**: the agent that ensures compliance with all the refined norms in the organization.
- **CommunicationAgent**: the agent responsible for controlling communication among agents, and for recording the interaction between agents and organizations.
- **Sniffer**: manages the message history and filters information by controlling communication initiated by queries.

In the Figure 1 the different agents of the system and the interactions among them are represented. In the upper corner of the figure we can see the task list that store the activities to carry out in the VO and in the center of the image, the agents and the interconnections are showed.

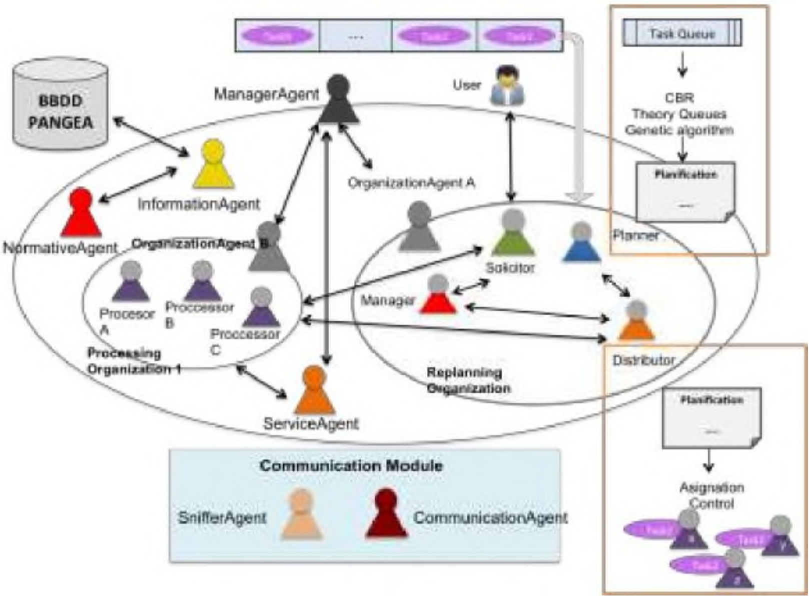

Fig. 1. Overview of the agent involved in the system

More information of PANGEA can be consulted in [14].

4 Results and Conclusions

PANGEA is a complete and innovative platform. We can conclude that PANGEA has great potential to create open multi-agent systems, and more specifically, virtual agent organizations. In order to assess the mechanism of the proposed planning, taking into account that the final goal is to process all the records within time limits or, if not possible, to maximize the benefit being carried out by different tests. The system was tested with two different records for four simulation Modes: Mode 1 – Without planning, Mode 2 - Calculating the number of agents needed within queuing theory, Mode 3 - Planning (including queuing theory and genetic algorithms) , Mode 4 – The whole planning with CBR. In the first case (Test 1) a list of 500 records were entered within a period of 120 minutes and the second case (Test 2) had 1500 records into the system within 210 minutes. Previously it had a memory of 700 cases, based on cases where values were altered randomly following a normal distribution.

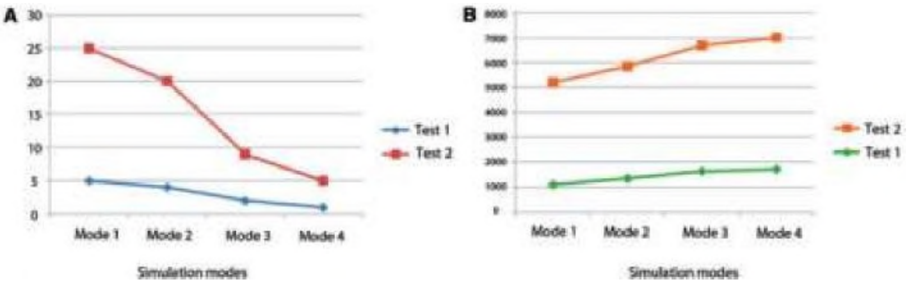

Fig 2. (A) Number of files out of time. (B) Profit earned

The left part of figure 2 shows the comparative number of cases that could not be processed in time for each mode in the two tests. The chart presented to the right in Figure 2 is even more significant because it reveals the benefits obtained in different modes. To assign a value to the benefit field for each record a point system is used taking into account various aspects such as the type of record, the economic gain expected, previous work, etc.

As expected, the worst benefit is obtained in an without planning mode. In this case, the fact that a generic estimate of the number of players is very vague so in many cases there are losses by not having enough agents or having too many. Also since it does not take into account the benefit of each record, it gives priority to some records which are considered losses. In the second case, the use of queuing theory improves the benefits being able to balance between the number of agents required and ordination benefit and thus assigning priority to improve the figures. In the third mode, as the above chart shows that the introduction of a task distribution based on genetic algorithm is the strong point of planning. Finally, still better benefits are obtained taking into account the adaptation from past cases provided by the CBR.

Acknowledgment. This work has been carried out by the project Sociedades Humano-Agente: Inmersion, Adaptacion y Simulacion. TIN2012-36586-C03-03. Ministerio de Economía y Competitividad. Fondos Feder.

References

1. Durfee, E.H., Lesser, V.R. and Corkill, D.D: Trends in Cooperative Distributed Problem Solving. IEEE Transactions on Knowledge and Data Engineering, Vol. 1 (1) 63-83 (1989)
2. Zambonelli, F., Jennings, N.R. and Wooldridge. M.: Developing multiagent systems: The Gaia methodology. ACMTransactions on Software Engineering and Methodology. Vol.12 (3) 317–370 (2003)
3. Esteva, M., Rodríguez, J., Sierra, C., Garcia, P., Arcos, J. On the formal specific ations of electronic institutions. Agent-mediated Electronic commerce. LNCS, 126–147 (2001)
4. Bajo, J., Corchado, J.J., Botti, V. and Ossowski, S. Practical Applications of Agents and MAS: Methods, Techniques and Tools for Open MAS. Journal of Physical Agents, Vol. 3 (2) (2009)
5. Kolodner, J. Case-Based Reasoning Morgan Kaufmann, San Francisco. (1993)
6. Corchado, E., Assumpcio M., MBorrajo, M.L., A maximum likelihood Hebbian learning-based method to an agent-based architecture. Int. Journal of Computer Mathematics. Vol.86 (10-11) 1760-1768 (2009)
7. Martín, Q. Investigación Operativa. Prentice-Hall. (2003)
8. Menasce, D.A. Trade-offs in designing Web clusters. Internet Computing, IEEE, Vol 6 (5) 76-80 (2002)
9. Shi, C., Lu, J. and Zhang, G.: An extended Kuhn-Tucker approach for linear bilevel programming. Applied Mathematics and Computation. Vol. 162 (1) 51-63 (2005)
10. Kabadi, S., and Punnen, A.: A strongly polynomial simplex method for the linear fractional assignment problem, Operations Research Letters. Vol. 36 (4) 402-407 (2008)
11. Wan, S., Yang, F. and Izquierdo, E.: Lagrange multiplier selection in wavelet-based scalable video coding for quality scalability, Signal Processing: Image Communication. Vol. 24 (9) 730-739 (2009)
12. Corchado, J.M., Gonzalez-Bedia, M., De Paz, Y., Bajo, J., De Paz, J. F.: Replanning mechanism for deliberative agents in dynamic changing environments. Computational Intelligence, 24 (2) (2008) 77-101
13. Zato, C., De Paz, J.F., De Luis, A., Bajo, J., Corchado, J.M.: Model for assigning roles automatically in egovernment virtual organizations, Expert Systems with Applications, 39 (12) (2012) 10389-10401.
14. Zato, C., Villarrubia, G., Sánchez, A., Bajo, J. and Corchado, J.M.: (2012) PANGEA: A New Platform for Developing Virtual Organizations of Agents. International Journal of Artificial Intelligence (IJAI), vol. 11, no. A13, pp. 93-102.