Multiagent Systems in Expression Analysis

Juan F. De Paz, Sara Rodríguez, and Javier Bajo

Departamento Informática y Automática
Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{fcofds, srg, jbajope}@usal.es

Abstract. This paper presents a multiagent system for decision support in the diagnosis of leukemia patients. The core of the system is a type of agent that integrates a novel strategy based on a case-based reasoning mechanism to classify leukemia patients. This agent is a variation of the CBP agents and proposes a new model of reasoning agent, where the complex processes are modeled as external services. The agents act as coordinators of Web services that implement the four stages of the case-based reasoning cycle. The multiagent system has been implemented in a real scenario, and the classification strategy includes a novel ESOINN neuronal network and statistics methods to analyze the patient's data. The results obtained are presented within this paper and demonstrate the effectiveness of the proposed agent model, as well as the appropriateness of using multiagent systems to resolve medical problems in a distributed way.

Keywords: Multiagent Systems, Case-Based Reasoning, microarray, neuronal network, ESOINN, Case-based planning.

1 Introduction

Currently, there exist many different systems aimed to provide decision support in medical environments [11] [12]. Cancer diagnosis is a field requiring novel automated solutions and tools, able to facilitate the early detection, even prediction, of cancerous patterns. The continuous growth of techniques for obtaining cancerous samples, specifically those using microarray technologies, provides a great amount of data. Microarray has become an essential tool in genomic research, making it possible to investigate global gene in all aspects of human disease [13]. Currently, there are several kinds of microarrays such as CGH arrays [16], expression arrays [17]. Expression arrays contain information about certain genes in patient's samples. Specifically, the HG U133 plus 2.0 [17] are chips used for this kind of analysis of expression. These chips analyze the expression level of over 47.000 transcripts and variants, including 38.500 well-characterized human genes. It is comprised of more than 54.000 probe sets and 1.300.000 distinct oligonucleotide feature. The great amount of data requiring analysis makes it necessary the use of data mining techniques in order to reduce the processing time. These data have a high dimensionality and require new powerful tools. Usually, the existing systems are focused on working with very concrete problems or diseases, with low dimensionality for the data, and it is very difficult to adapt them to new contexts for diagnosis of different diseases. Nowadays, there are

Y. Demazeau et al. (Eds.): 7th International Conference on PAAMS'09, AISC 55, pp. 217–226.
springerlink.com

different approximations as myGrid [23] [20] aimed to resolve this situation. They base their functionality in the creation of web service that are implemented following the OGSA (Open Grid Services Architecture) [14], but the main disadvantage is that the user must be the responsible of creating the sequence of actions that resolve a concrete problems. These systems provide methods for resolving complex problems in a distributed way through SOA [19] and Grid architectures, but lack of capacity for adaptation. On the other hand, there exist new research lines focused on reasoning mechanism with high capacity for learning and adaptation. Among these mechanisms highlights the case-based Reasoning (CBR) systems [2], which solve new problems taking into account the knowledge obtained in previous experiences [2], [15] and its integration within agents and multiagent systems. The inconvenience of the decision support systems based on CBR for microarray classification is the high dimensionality of the data and the corresponding complexity.

One alternative to the SOA architectures are multi-agent systems [18], that provide distributed entities with autonomous reasoning skills, called agents. Some proposals provide the agents with special capabilities for learning and adaptation by means of CBP (Case-Based Planning) mechanisms [3]. However, the CBP-BDI agents [3] present lacks when working with problems of high dimensionality and their efficiency is reduced. In [24] the incorporation of web services in multiagent architectures for implementing the agent's capabilities is studied. MAS and SOA architectures have been integrated and used in fields as gas turbine plant control [25] or hydrocarbure industries [26] to define the processes workflow.

This paper presents an innovative solution to model decision support Systems consisting of a multi-agent architecture which allows integration with Web services. In this sense it is possible to analyse data, for example from microarrays, in a distributed way. Moreover, the architecture incorporates CBP (Case-based planning) agents [3] specifically designed to act as coordinators of Web services. Thus, it is possible to reduce the computational load for the agents and expediting the classification process. The DASA architecture proposed within this paper has been applied to a case study, consisting of the classification of leukemia patients, and incorporates novel strategies for data analysis and classification. The process of studying a microarray is called expression analysis [1] and consists of a series of phases: data collection, data preprocessing, statistical analysis, and biological interpretation. These phases analysis consists basically of three stages: normalization and filtering; clustering and classification; and extraction of knowledge. In this work, a multiagent system based on the DASA architecture models the phases of the expression analysis by means of agents and incorporates innovative algorithms implemented as Web services, as filtering techniques based on statistical analysis, allowing a notable reduction of the data dimensionality and a classification technique based on a ESOINN [4] neural network. The core of the system are reasoning agents based on the CBP [3] mechanism. Furthermore, the system incorporates other agent types to accomplish complementary tasks required for the expression analysis.

Section 2 presents the DASA architecture. Section 3 describe a case study and finally,Section 4 presents the results and conclusions obtained.

2 DASA Architecture

DASA (Device, Agents and Service Architecture) is a multi-agent architecture that incorporates agents with skills to generate plans for analysis of large amounts of data. This is a novel mechanism for the implementation of the stages of CBP mechanisms through Web services. The architecture provides communication mechanisms that facilitate integration with SOA architectures.

DASA has been initially designed to facilitate the processing of data from expression arrays. To do this, DASA has been divided into three main blocks: devices modules, agents and services. The services are responsible for carrying out the processing of information by providing replication features and modularity. The agents act as coordinators and managers of services. Agents in the organization layer are available to run on different types of devices, so are created versions suitable to them.

The agents layer constitutes the core of the architecture, as can be seen in Figure 1. Figure 1 shows four groups of agents:

- Organization: The agents of the organization run on the user devices or on servers. The agents installed on the devices of the users make a bridge between the devices and agents of the system that perform data analysis. The agents installed on servers will be responsible for conducting the analysis of information on the model of reasoning CBP [3].
- Analysis: The agents in the analysis layer are responsible for selecting the configuration of the services that better suit the problem to solve. They communicate with Web services to generate results. The agents of this layer follow the model of reasoning CBP [3].
- Representation: These agents are in charge of generating the tables with the classification data and the graphics for the results.
- Import/Export: These agents are in charge of formatting the data in order to adjust them to the needs of agents and services.
- The Controller agent module manages the agents available in the analysis layer. It allows the registration of agents in the layer, as well as their use in the organization.

On the other hand, the services layer is divided into two groups:

- Analysis Services: The analysis services are services used by agents of analysis for carrying out different tasks. Within the analysis services are services for pre-processing, filtering, clustering and extraction of knowledge.
- Representation Services: They generate graphics and result tables.

Within the services layer, there is a service called Facilitator Directory that provides information on the various services available and manages the XML file for the UDDI (Universal Description Discovery and Integration). To facilitate communication between agents and services the architecture integrates a communication layer that provides support for the FIPA-ACL and SOAP protocols.

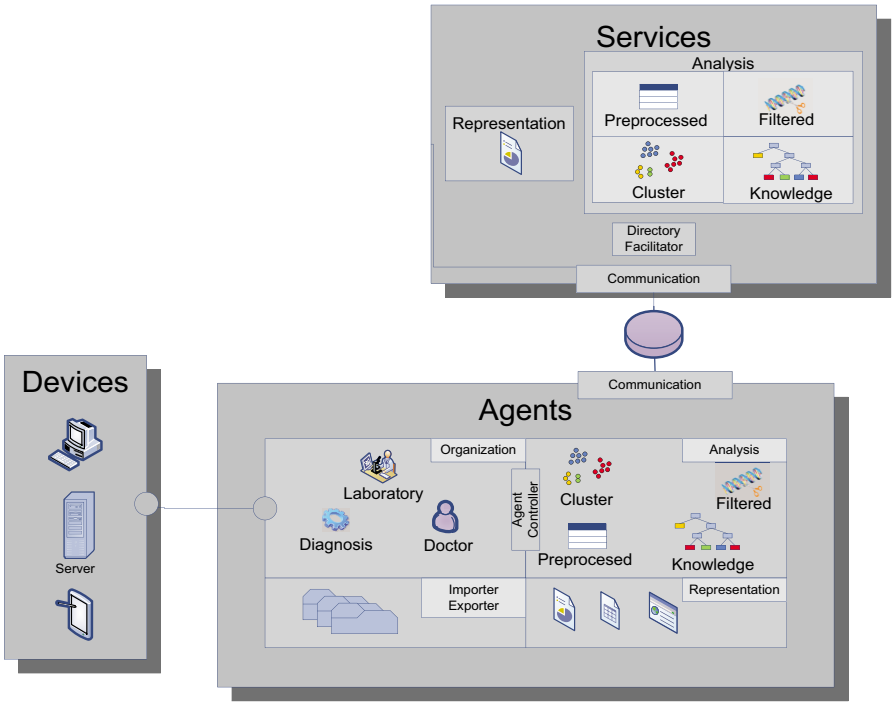

Fig. 1. Arquitectura de DASA

2.1 Coordinator Agent Based on CBR and CBP

The agents in the organization layer and the agents in the analysis layer have the capacity to learn from the analysis carried out in previous procedures. To do so, they adopt the model of reasoning CBP, a specialization of case-based reasoning (CBR) [2]. The primary concept when working with CBP's is the concept of case. A case can be defined as a past experience, and is composed of three elements: A problem description, which describes the initial problem; a solution, which provides the sequence of actions carried out in order to solve the problem; and the final state, which describes the state achieved once the solution was applied. A CBR manages cases (past experiences) to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four sequential phases: retrieve, reuse, revise and retain.

Case-based planning (CBP) is the idea of planning as remembering [3]. In CBP, the solution proposed to solve a given problem is a plan, so this solution is generated taking into account the plans applied to solve similar problems in the past. The problems and their corresponding plans are stored in a plans memory.

A plan P is a tuple $<S,B,O,L>$:

- S is the set of plan actions.
- O is an ordering relation on S allowing to establish an order between the plan actions.

- *B* is a set that allows describing the bindings and forbidden bindings on the variables appearing in *P*.
- *L* is a set of casual links. That is, relations allowing to establish a link between plan actions.

3 Case Study: Decision Support for Leukemia Patients Diagnosis

The DASA multiagent architecture has been used to Developer a decision support system for the classification of leukemia patients. In framework of this research the system developed had available 212 samples from analyses performed on patients either through punctures in marrow or blood samples and affected by five types of leukemia (ALL, CLL, AML, MDS, CML). In this way, the analysis of data from microarrays is made in a distributed manner and certain tedious tasks are automated. The process consists of three steps: Initially the laboratory personnel hybridizes the samples, then the data analysis is made and finally a human expert interprets the results obtained. The multiagent system built from the DASA architecture reproduces this behaviour. The aim of the tests performed is to determine whether the system is able to classify new patients based on the previous cases analyzed and stored. Next, the developed agents and services are explained.

3.1 Services Layer

The services implement the algorithms that allow the analysis expression of the microarrays [1]. These services are invoked by the agents and present novel analysis techniques.

3.1.1 Preprocessing Service

This service implements the RMA algorithm and a novel control and errors technique. The RMA (R*obust Multi-array Average*) [5] algorithm is frequently used for preprocessing Affymetrix microarray data. RMA consists of three steps: (i) Background Correction; (ii) Quantile Normalization (the goal of which is to make the distribution of probe intensities the same for arrays); and (iii) Expression Calculation. During the Control and Errors phase, all probes used for testing hybridization are eliminated. Some few control points should contain the same values for all individuals. On occasion, some of the measures made during hybridization may be erroneous; not so with the control variables. In this case, the erroneous probes that were marked during the RMA must be eliminated.

3.1.2 Filtering Service

The filtering service eliminates the variables that do not allow classification of patients by reducing the dimensionality of the data. Three services are used for filtering: **Variability:** Las variables con baja variabilidad no poseen valores similares para todos los individuos por tanto no son significativas a la hora de realizar clasificaciones. The first stage is to remove the probes that have low variability according to the following steps: Calculate the standard deviation for each of the probes, standardize the

above values, discard of probes for which the value of z meet the following condition: $z < \alpha$. **Uniform Distribution:** All remaining variables that follow a uniform distribution are eliminated. The variables that follow a uniform distribution will not allow the separation of individuals. The contrast of assumptions followed is explained below, using the Kolmogorov-Smirnov [6] test. H_0: The data follow a uniform distribution; H_1: The analyzed data do not follow a uniform distribution. **Correlations:** The linear correlation index of Pearson is calculated and correlated variables are removed so that only the independent variables remain.

3.1.3 Clustering Service

It addresses both the clustering and the association of a new individual to the group more appropriate. The services included in this layer are: the ESOINN neural network [4] (Enhanced self-organizing incremental neuronal network) y the NN clustering algorithm (Nearest Neighbor). Additional services in this layer for clustering are the Partition around medoids (PAM) [21] and demdograms [22].

The ESOINN [4] (Enhanced self-organizing incremental neuronal network) clustering technique is variation of neural network SOINN (self-organizing incremental neuronal network) [7]. ESOINN consists of a single layer, so it is not necessary to determine the manner in which the training of the first layer changes to the second. With a single layer, ESOINN is able to incorporate both the distribution process along the surface and the separation between low density groups. The initial phase could be understood as a phase of competition, while in a second phase, the network of nodes begins to expand just as with a NG. The classification is carried out bearing in mind the similarity of the new case using the NN cluster. The similarity measure used is as follows:

$$d(n,m) = \sum_{i=1}^{s} f(x_{ni}, x_{mi}) * w_i \tag{1}$$

Where s is the total number variables, n and m the cases, w_i the value obtained in the uniform test and f the Minkowski [8] Distance that is given for the following equation.

$$f(x,y) = \sqrt[p]{\sum_i \left| x_i - y_j \right|^p} \quad \text{con } x_i, y_j \in R^p \tag{2}$$

This dissimilarity measure weighs those probes that have a less uniform distribution, since these variables don't allow a separation.

3.1.4 Knowledge Extraction Service

The extraction of knowledge technique applied has been CART (Classification and Regression Tree) [9] algorithm is carried out. The CART algorithm is a non parametric test that allows extracting rules to explain the classification carried out. There are others techniques to generate the decision trees, as the methods based on ID3 trees [10], although the most used currently is CART.

3.2 Agents Layer

The agents in the analysis layer implement the CBP reasoning model and, for this, select the flow for services delivery and decide the value of different parameters based on previous plans made. A measure of efficiency is defined for each of the agents to determine the best course of the recovered for each phase of the analysis process.

In the analysis layer, at the stage Preprocessed only a service is available, so that the agent only selects the settings. The efficiency is calculated by the deviation in the microarray once have been preprocessed. All the cases are recovered and the configuration with greater efficiency is selected. At the stage of filtering, the efficiency of the plan p is calculated by the relationship between the proportion of probes and the resulting proportion of individuals falling ill.

$$e(p) = \frac{s}{N} + \frac{i'}{I} \tag{3}$$

Where s is the final number of variables, N is the initial number of probes, i' the number of individuals misclassified and I the total number of individuals. In the phase of clustering and classification the efficiency is determined by the number of individuals misclassified. The measure of similarity to retrieve the most similar plans is defined as the difference in the number of probes. Finally, in the process of extracting knowledge at the moment, CART has only been implemented for this task so that the agent responsible for conducting the selection method does not take into account the plans recovered, as in the previous phase Efficiency is determined by the number of individuals misclassified.

In the organization layer, the diagnosis agent is in charge of choosing the agents for the expression analysis [1]. The diagnosis agent establishes the number of plans to recover from the plans memory for each of the agents as well as the agents to select from the analysis layer. In a similar way the laboratory agent and the human expert select the agents from the organization layer that will be used. If the review at all stages is positive, and the human expert feels good result, the plan is stored in the memory of plans for further use.

4 Results and Conclusions

This paper has presented the DASA multiagent architecture and its application to a real problem. DASA facilitates tasks automation by means of intelligent agents capable of autonomously plan the stages of an expression analysis. Moreover, DASA facilitates the distributed execution of complex computational services, reducing the number of crashes in agents, since DASA separates the processing tasks from the agent architecture, and consequently, reduces the posibility of failure due to agents overload [27]. The multiagent system developed integrates within web services aimed to reduce the dimensionality of the original data set and a novel method of clustering for classifying patients. The multiagent perspective allow the system to works in a way similar to how human specialists operate in the laboratory, but is able to work with great amounts of data and make decisions automatically, thus reducing significantly both the time required to make a prediction, and the rate of human error due to confusion. The system focuses on identifying the important variables for each of the variants of blood cancer so that patients can be classified according to these variables.

Table 1. Plans of the filtering phase and plan of greater efficiency

Variability (z)	Uniform (α)	Correlation (α)	Probes	Errors	Efficiency
-1.0	0.25	0.95	2675	21	0.1485
-1.0	0.15	0.90	1341	23	0.1333
-1.0	0.15	0.95	1373	24	0.1386
-0.5	0.15	0.90	1263	24	0.1365
-0.5	0.15	0.95	1340	23	0.1333
-1.0	**0.1**	**0.95**	**785**	**24**	**0.1277**
-1.0	0.05	0.90	353	32	0.1574
-1.0	0.05	0.95	357	34	0.1669
-0.5	0.05	0.9	332	47	0.2278
-0.5	0.05	0.95	337	53	0.2562
-1.0	0.01	0.95	54	76	0.3594

In the study of leukaemia on the basis of data from microarrays, the process of filtering data acquires special importance. In the experiments reported in this paper, we worked with a database of bone marrow cases from 212 adult patients with five types of leukaemia. Table 1 shows the plans managed by the filtering agent for the analysis of the data from the HG U133 chip. Table 1 shows the values of the different parameters, significance levels for the tests and efficiency, and has been generated from the previous analysis of the 212 individuals.

Subsequently, we proceeded to repeat the test without pre-established settings, so that the agent automatically selected the plan of greater efficiency from equation (3). It has been highlighted in bold plan selected.

In Figure 2 it is possible to observe the performance of DASA for each of the agents of the organization and analysis layers. 11 tests were conducted based on manual planning and the results were compared with the automatic analysis provided by the multiagent system. Each of the agents of the organization layer selects the agents from the analysis layer as the previous plans and, in turn, each of these agents selects the services and configuration parameters. At the bottom of Figure 2 it can be seen the kind of agent of the analysis layer, and at the top the agent of the organization layer.

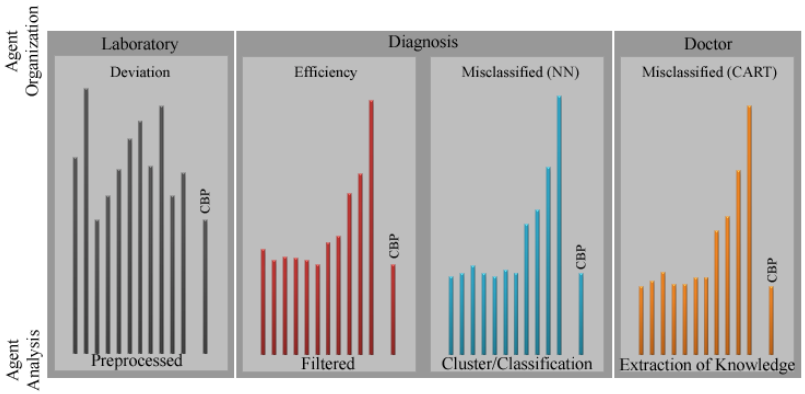

Fig. 2. DASA Architecture

In each chart the efficiency measure used is shown. The bar for the CBP agent is the highest efficiency according to the definitions applied.

DASA distributes the functionality among Web services, automatically calculates the expression analysis and allows the classification of leukaemia patients from the microarray data. DASA notably improves the performance provided by the manual procedures.

Acknowledgements. Special thanks to the Institute of Cancer of Salamanca for the information and technology provided. This work was supported in part by the projects MEC THOMAS TIN2006-14630-C03-03, IMSERSO 137/07 and JCYL SA071A08.

References

1. Lander, E., et al.: Initial sequencing and analysis of the human genome. Nature 409, 860–921 (2001)
2. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Francisco (1993)
3. Glez-Bedia, M., Corchado, J.: A planning strategy based on variational calculus for deliberative agents. Computing and Information Systems Journal 10(1), 2–14 (2002)
4. Furao, S., Ogura, T., Hasegawa: An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks 20, 893–903 (2007)
5. Irizarry, R., Hobbs, B., Collin, F., Beazer-Barclay, Y., Antonellis, K., Scherf, U., Speed, T.: Exploration, Normalization, and Summaries of High density Oligonucleotide Array Probe Level Data. Biostatistics 4, 249–264 (2003)
6. Brunelli, R.: Histogram Analysis for Image Retrieval. Pattern Recognition 34, 1625–1637 (2001)
7. Shen, F.: An algorithm for incremental unsupervised learning and topology representation. Ph.D. thesis. Tokyo Institute of Technology, Tokyo (2006)
8. Gariepy, R., Pepe, W.: On the Level sets of a Distance Function in a Minkowski Space. Proceedings of the American Mathematical Society 31(1), 255–259 (1972)
9. Breiman, L., Friedman, J., Olshen, A., Stone, C.: Classification and regression trees. In: Wadsworth International Group, Belmont, California (1984)
10. Quinlan, J.: Discovering rules by induction from large collections of examples. In: Michie, D. (ed.) Expert systems in the micro electronic age, pp. 168–201. Edinburgh University Press, Edinburgh (1979)
11. Chua, A., Ahna, H., Halwanb, B., Kalminc, B., Artifond, E., Barkune, A., Lagoudakisf, M., Kumar, A.: A decision support system to facilitate management of patients with acute gastrointestinal bleeding. Artificial Intelligence in Medicine 42(3), 247–259 (2008)
12. François, P., Cremilleux, B., Robert, C., Demongeot, J.: MENINGE: A medical consulting system for child's meningitis. Study on a series of consecutive cases. Artificial Intelligence in Medicine. 32(2), 281–292 (1992)
13. Quackenbush, J.: Computational analysis of microarray data. Nature Review Genetics 2(6), 418–427 (2001)
14. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology Of The Grid: An Open Grid Services Architecture For Distributed Systems Integration. Technical Report of the Global Grid Forum (2002)
15. Leake, D., Kendall-Morwick, J.: Towards case-based support for e-science workflow generation by mining provenance. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 269–283. Springer, Heidelberg (2008)

16. Shinawi1, M., Cheung, S.W.: The array CGHnext term and its clinical applications. Drug Discovery Today 13(17-18), 760–770 (2008)
17. Affymetrix, http://www.affymetrix.com/support/technical/datasheets/hgu133arrays_datasheet.pdf
18. Wooldridge, M., Jennings, N.: Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge, Jennings (eds.) Intelligent Agents, pp. 1–22. Springer, Berlin (1995)
19. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR, Englewood Cliffs (2005)
20. Vittorini, P., Michettia, M., di Orio, F.: A SOA statistical engine for biomedical data. Computer Methods and Programs in Biomedicine 92(1), 144–153 (2008)
21. Saitou, N., Nie, M.: The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4, 406–425 (1987)
22. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics (1990)
23. Stevens, R., McEntireb, G.C., Greenwooda, M., Zhaoa, J., Wipatc, A., Lic, P.: MyGrid and the drug discovery process. Drug Discovery Today: BIOSILICO 2(4), 140–148 (2004)
24. Huhns, M., Singh, M.P.: Service-Oriented Computing: Key Concepts and Principles. Internet Computing 9(1), 75–81 (2005)
25. Arranz, A., Cruz, A., Sanz-Bobi, M.A., Ruíz, P., Coutiño, J.: DADICC: Intelligent system for anomaly detection in a combined cycle gas turbine plant. Expert Systems with Applications 34(4), 2267–2277 (2008)
26. Contreras, M., Sheremetov, L.: Industrial application integration using the unification approach to agent-enabled semantic SOA. Robotics and Computer-Integrated Manufacturing 24(5), 680–695 (2008)
27. Tapia, D.I., Rodriguez, S., Bajo, J., Corchado, J.M.: FUSION@, A SOA-Based Multiagent Architecture. In: International Symposium on Distributed Computing and Artificial Intelligence, Advances in Soft Computing, vol. 50, pp. 99–107 (2008)