MACSDE: Multi-Agent Contingency Response System for Dynamic Environments

Aitor Mata[1], Belén Pérez[1], Angélica González[1], Bruno Baruque[2] and Emilio Corchado[2]

[1] University of Salamanca, Spain
[2] University of Burgos, Spain
aitor@usal.es, lancho@usal.es, angelica@usal.es, bbaruque@ubu.es, escorchado@ubu.es

Abstract. Dynamic environments represent a quite complex domain, where the information available changes continuously. In this paper, a contingency response system for dynamic environments called MACSDE is presented. The explained system allows the introduction of information, the monitoring of the process and the generation of predictions. The system makes use of a Case-Based Reasoning system which generates predictions using previously gathered information. It employs a distributed multi-agent architecture so that the main components of the system can be accessed remotely. Therefore, all functionalities can communicate in a distributed way, even from mobile devices. The core of the system is a group of deliberative agents acting as controllers and administrators for all functionalities. The system explained includes a novel network for data classification and retrieval. Such network works as a summarization algorithm for the results of an ensemble of Self-Organizing Maps. The presented system has been tested with data related with oil spills and forest fire, obtaining quite hopeful results.

Keywords: Dynamic environments; Case-Based Reasoning; oil spill; forest fire; Self Organizing Memory; summarization.

1 Introduction

Dynamic systems represent a quite difficult knowledge field where new techniques and architectures can be applied to check their useful applications. In this paper, a multi-agent architecture is applied to natural dynamic systems, which use a large number of parameters and contain a high level of system variability.

This paper presents the MACSDE architecture. It is a new approach to open systems, where the different parts of the applications that are implemented communicate with each other by an internal architecture that allows different elements to connect to the system or to eventually disconnect from the system if they are not going to be operative for a time.

The system explained in this work implements the Case-Based Reasoning methodology as the core of the system, and generated the predictions for solving the problems presented. The current system has been expanded from a previously local version to one that is distributed, where different users with different roles communicate with each other through the MACSDE architecture, and use the CBR inner methodology to solve problems and generate predictions. The main phases of the CBR cycle are implemented in this application since services are already part of the MACSDE structure.

This paper presents the problems that the architecture faces with regards to predictions in natural environments specifically oil spills and forest fires. In both cases, the system must generate predictions in order to know, first of all, where the oil slicks that generated after a spill will be, and secondly, where the fire will be at a specific time.

The remainder of this paper is structured as follows: section 2 provides a brief explanation of the two basic methodologies used in MACSDE; section 3 presents the proposed model as well as a description of the services offered; section 4 shows the results obtained with MACSDE, being applied to oil spills and forest fires, and finally section 5 includes the conclusions obtained and future lines of work.

2 Combining a multi-agent architecture and case-based reasoning systems

Agents and multi-agent systems have been successfully applied to several scenarios, such as education, culture, entertainment, medicine, robotics, etc. [1]. Agents have a set of characteristics, such as autonomy, reasoning, reactivity, social abilities, pro-activity, mobility, organization, etc. which allow them to cover several needs for developing contingency response systems [2].

The agents' characteristics make them appropriate for developing dynamic and distributed systems, as they possess the capability of adapting themselves to the users and environmental characteristics [3]. In addition, the continuous advances in mobile computing make it possible to obtain information about the environment and also to react physically over it in more innovative ways. The agents in MACSDE multi-agent system are based on the deliberative (Belief, Desire, Intention - BDI) model [4], where the agents' internal structure and capabilities are based on mental aptitudes, using beliefs, desires and intentions for solving problems. However, modern developments need higher adaptation, learning and autonomy levels than pure BDI model [4]. This can be achieved by modeling the agents' characteristics for providing them with mechanisms that allow them solving complex problems and achieving autonomous learning. Some of these mechanisms are Case-Based Reasoning (CBR) systems [5], where problems are solved by using solutions to similar past problems [1]. The origin of the CBR methodology is in knowledge based systems and has been used on several scenarios such as health care, e-Learning, ubiquitous computing, oceanography, etc. [6]. Solutions are stored into a case memory which the system can consult in order to find better solutions for new problems. Deliberative agents can make use of these systems to learn from past experiences and to adapt their behavior according each situation.

The main element of CBR is the case base which is a structure that stores problems, elements (cases) and its solutions. A case base can be visualized as a database where a collection of problems is stored keeping a relationship with the solutions to every problem stored. This gives the system the ability to generalize in order to solve new problems. In the case of the present work the case base is implemented by means of a topology preserving meta-algorithm called Weighted Voting Superposition of Self-Organizing Maps (WeVoS-SOM). The learning capabilities of CBR are due to its own structure composed of four main phases [5]: retrieval, reuse, revision and retention.

The retrieve phase consists on finding the most similar cases to the current problem from the case base. Once a series of cases are extracted they can be reused. The reuse phase adapts the selected cases for solving the current problem through a new solution. The revision phase revises the solution to check if it is an adequate solution to the current problem. If the solution is confirmed, then it is retained by means of the retain phase. This new solution could eventually serve as a solution for future problems. In most cases, CBR should not been used alone but combined with artificial intelligence techniques on each phase. For example, Growing Cell Structures has been used with CBR to automatically create the intern structure of the case base from existing data and it has been combined with multi-agent applications [7] to improve its results. ART-Kohonen neural networks, artificial neural networks, genetic algorithms and fuzzy logic [8] have been used to complement the capabilities of CBR systems. These techniques enhance the way CBR systems create better solutions.

3 Presenting a new multi-agent contingency response system for dynamic environments

Cases are the key to obtain solutions to future problems through a CBR system. The functionalities of MACSDE (Multi-Agent Contingency response System for Dynamic Environments) can be accessed using different interfaces executed on PCs or PDAs (Personal Digital Assistant). Users can interact with the system by introducing data, requesting a prediction or revising a solution generated (i.e. prediction). The interface agents communicate with the services through the agents' platform and vice versa. The interface agents perform all the different functionalities which users can make use for interacting with MACSDE. The different phases of the CBR system have been modeled as services, so each phase can be requested independently. For example, one user may only introduce information in the system (e.g. a new case), while another user could request a new prediction.

All information is stored in the case base and MACSDE is ready to predict future situations. A problem situation must be introduced in the system for generating a prediction. Then, the most similar cases to the current situation are retrieved from the case base. Once a collection of cases are chosen from the case base, they must be used for generating a new solution to the current problem. Growing Radial Basis Functions (RBF) Networks [9] are used in MACSDE for combining the chosen cases in order to obtain the new solution.

MACSDE determines future probabilities in a certain area. It divides the area to be analyzed in squares of approximately half a degree side for generating a new prediction. Then, the system determines the amount of slicks in each square. The squares are colored with different gradation depending on the probability of finding the analyzed element.

Fig. 1 shows the structure of MACSDE, where the different elements of the system are related and where the communication paths between the elements are also shown.

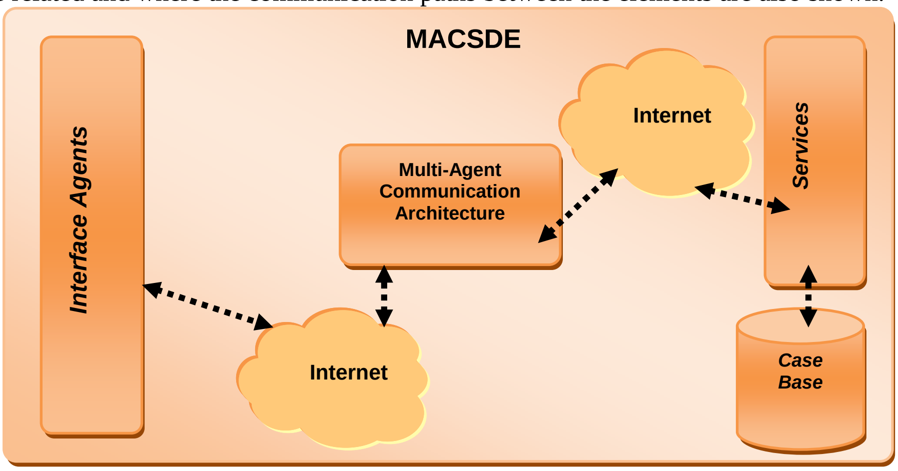

Fig. 1. MACSDE Architecture.

There are four basic blocks in MACSDE: Applications, Services, Agent Platform and Communication Protocol. These blocks provide all the system functionalities:

Interface Agents. These represent all the programs that users can use to exploit the system functionalities. Applications are dynamic, reacting differently according to the particular situations and the services invoked. They can be executed locally or remotely, even on mobile devices with limited processing capabilities, because computing tasks are largely delegated to the agents and services.

Services. These represent the activities that the architecture offers. They are the bulk of the functionalities of the system at the processing, delivery and information acquisition levels. Services are designed to be invoked locally or remotely. Services can be organized as local services, web services, GRID services, or even as individual stand alone services. MACSDE has a flexible and scalable directory of services, so they can be invoked, modified, added, or eliminated dynamically and on demand. It is absolutely necessary that all services follow a communication protocol to interact with the rest of the components. Services make use of the information stored in the case base to solve the proposed problems.

Multi-Agent Communication Architecture. This is the core of the system, integrating a set of agents, each one with special characteristics and behavior. An important feature in this architecture is that the agents act as controllers and administrators for all applications and services, managing the adequate functioning of the system, from services, applications, communication and performance to reasoning and decision-making. In MACSDE, services are managed and coordinated by deliberative BDI agents. The agents modify their behavior according to the users' preferences, the knowledge acquired from previous interactions, as well as the choices available to respond to a given situation. The communication protocol allows applications and services to communicate directly with the Agents Platform. This protocol is based on SOAP specification to capture all messages between the platform and the services and applications [10]. Services and applications communicate with the *Agents Platform* via SOAP messages. A response is sent back to the specific service or application that made the request. All external communications follow the same protocol, while the communication among agents in the platform follows the FIPA Agent Communication Language (ACL) specification.

MACSDE also defines three different services which perform all tasks that the users may demand from the system. All requests and responses are handled by the agents. The requests are analyzed and the specified services are invoked either locally or remotely. Services process the requests and execute the specified tasks. Then, services send back a response with the result of the specific task. In this way, the agents act as interpreters between applications and services in MACSDE. Next, CBR system used in MACSDE is explained.

3.1. Data Input Service

Case-Based Reasoning systems are highly dependent on stored information. The novel algorithm presented here, *Weighted Voting Summarization of SOM ensembles (WeVoS-SOM)* [11, 12] is used to organize the data that is accumulated in the case base. It is also used to recover the most similar cases to the proposed problem.

The main objective of the WeVoS-SOM is to generate a final map processing several other similar maps unit by unit. Instead of trying to obtain the best position for the units of a single map trained over a single dataset, it aims to generate several maps over different parts of the dataset. Then, it obtains a final summarized map by calculating by consensus which is the best set of characteristics vector for each unit position in the map. To do this calculation, first this meta-algorithm must obtain the "quality" [13] of every unit that composes each map, so that it can relay in some kind of informed resolution for the fusion of neurons.

The final map obtained is generated unit by unit. The units of the final map are first initialized by determining their centroids in the same position of the map grid in each of the trained maps. Afterwards, the final position of that unit is recalculated using data related with the unit in that same position in every of the maps of the ensemble. For each unit, a sort of voting process is carried out as shown in Eq. 1:

$$V_{p,m} = \frac{\sum b_{pm}}{\sum_1^M b_p} \cdot \frac{q_{pm}}{\sum_1^M q_p} \qquad (\mathbf{1})$$

where, V_{pm} is the weight of the vote for the unit included in map m of the ensemble, in its position p; M is the total number of maps in the ensemble; b_{pm} is the binary vector used for marking the dataset entries recognized by the unit in position p of map m; and, q_{pm} is the value of the desired quality measure for the unit in position p of map m.

The final map is fed with the weights of the units as it is done with data inputs during the training phase of a SOM [14], considering the "homologous" unit in the final map as the BMU. The weights of the final unit will be updated towards the weights of the composing unit. The difference of the updating performed for each "homologous" unit in the composing maps depends on the quality measure calculated for each unit. The higher the quality (or the lowest error) of the unit of the composing map, the stronger the unit of the summary map will be updated towards the weights of that unit. The summarization algorithm will consider the weights of a composing unit "more suitable" to be the weights of the unit in the final map according to both the number of inputs recognized and the quality of adaptation of the unit (Eq. 1). With this new approach it is expected to obtain more faithful maps to the inner structure of the dataset.

3.2. Prediction Generation Service

When a prediction is requested by a user, the system starts recovering from the case base the most similar cases to the problem proposed. Then, it creates a prediction using artificial neural networks. Once the most similar cases are recovered from the case base, they are used to generate the solution. Growing RBF networks [15] are used to obtain the predicted future values corresponding to the proposed problem. This adaptation of the RBF networks allows the system to grow during training gradually increasing the number of elements (prototypes) which play the role of the centers of the radial basis functions. The creation of the Growing RBF must be made automatically which implies an adaptation of the original GRBF system. The error for every pattern is defined by (Eq. 2).

$$e = {}^{l}\!/_{n} \sum_{k=1}^{p} \left\| t_{ik} - y_{ik} \right\| \tag{2}$$

where t_{ik} is the desired value of the k_{th} output unit of the i_{th} training pattern, y_{ik} the actual values of the k_{th} output unit of the i_{th} training pattern.

Once the GRBF network is created, it is used to generate the solution to the proposed problem. The solution proposed is the output of the GRBF network created with the retrieved cases. The GRBF network receives, as input, the values stored in the case base. With those values, the network generates the proposed solution, using only the data recovered from the case base in previous phases.

3.3. Revision Service

After generating a prediction, the system needs to validate its correction. MACSDE can also query an expert user to confirm the automatic revision previously done. The system also provides an automatic method of revision that must be also checked by an expert user which confirms the automatic revision.

Explanations are a recent revision methodology used to check the correction of the solutions proposed by CBR systems [16]. Explanations are a kind of justification of the solution generated by the system. To obtain a justification to the given solution, the cases selected from the case base are used again. As explained before, a relationship between a case and its future situation can be established. If both the situations defined by a case and the future situation of that case are considered as two vectors, a distance between them can be defined, calculating the evolution of the situation in the considered conditions. That distance is calculated for all the cases retrieved from the case base as similar to the problem to be solved. If the distance between the proposed problem and the solution given is not greater than the average distances obtained from the selected cases, then the solution is a good one, according to the structure of the case base. If the proposed prediction is accepted, it is considered as a good solution to the problem and can be stored in the case base in order to solve new problems. It will have the same category as the historical data previously stored in the system.

4 Results

MACSDE has been tested in two different fields, both related with natural dynamic environments. It has been checked with a subset of the available data that has not been previously used in the training phase. The predicted situation was contrasted with the actual future situation as it was known (historical data was used to train the system and also to test its correction). The proposed solution was, in most of the variables, close to 90% of accuracy.

Table 1. Percentage of good predictions obtained with different techniques.

Number of cases	RBF		CBR		RBF + CBR		MACSDE	
	Oil spill	*Forest fires*	*Oil spill*	*Forest fires*	*Oil spill*	*Forest fires*	*Oil spill*	*Forest fires*
500	43 %	38 %	40 %	41 %	43 %	44 %	46 %	47 %
1000	47 %	44 %	48 %	47 %	52 %	51 %	62 %	66 %
3000	54 %	51 %	56 %	53 %	64 %	65 %	78 %	73 %
5000	61 %	58 %	63 %	61 %	74 %	70 %	87 %	84 %

Table 1 shows a summary of the obtained results. In this table different techniques are compared. The evolution of the results is shown along with the augmentation of the number of cases stored in the case base. All the techniques analyzed improve their results at the same time the number of stored cases is increased. The solution proposed do not generate a trajectory, but a series of probabilities in different areas, what is far more similar to the real behaviour of the oil slicks. The left column of each technique is referred to the oil spill problem, while the right column is referred to the forest fires.

The "*RBF*" column represents a simple Radial Basis Function Network that is trained with all the data available. The network gives an output that is considered a solution to the problem. The "*CBR*" column represents a pure CBR system, with no additional techniques included. The cases are stored in the case bases and recovered considering the Euclidean distance. The most similar cases are selected and after applying a weighted mean depending on the similarity, a solution is proposed. It is a *mathematical* CBR. The "*RBF + CBR*" column corresponds to the possibility of using a RBF system combined with CBR. The recovery from the CBR is done using the Manhattan distance to determine the closest cases to the introduced problem. The RBF network works in the reuse phase, adapting the selected cases to obtain the new solution. The results of the "*RBF+CBR*" column are, normally, better than those of the "*CBR*", mainly because of the elimination of useless data to generate the solution. Finally, the "*MACSDE*" column shows the results obtained by the proposed system, being better than the three previous solutions analyzed.

Several tests have been done to compare the overall performance of MACSDE. The tests consisted of a set of requests delivered to the Prediction Generation Service (PGS) which in turn had to generate solutions for each problem. There were 50 different data sets, each one with 10 different parameters. The data sets were introduced into the PGS through a remote PC running multiple instances of the Prediction Agent. The data sets were divided in five test groups with 1, 5, 10, 20 and 50 data sets respectively. There was one Prediction Agent for each test group. 30 runs for each test group were performed. First, all tests were performed with only one Prediction Service running in the same workstation on which the system was running. Then, five Prediction Services were replicated also in the same workstation. For every new test, the case base of the PGS was deleted in order to avoid a learning capability, thus requiring the service to accomplish the entire prediction process.

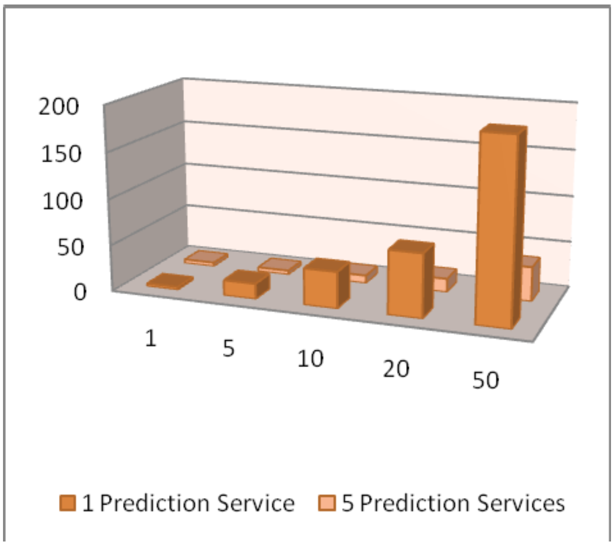

Fig. 2. Average time needed to generate all solutions.

Fig. 2 shows the average time needed by MACSDE for generating all solutions for each test group. The time exponentially increases when there is only one PGS running. This is because the service must finish a request to start the next one. So, for the last test group (50 data sets) the service was overcharged. On the other hand, with five replicated services, the system can distribute the requests among these services and optimize the overall performance. The system performed slightly faster when processing a single request, but the performance was constantly reduced when more requests were sent to the service.

5 Conclusions and Future Work

As conclusions, a novel hybrid model called MACSDE, which aim is to provide the users with predictions to asses their actions regarding the contingency response in dynamic environments, is presented in this work.

It has been tested under the frame of two different real cases and compared with other previous models. From this comparison it can be concluded that the novel hybrid model presented outperforms previous simpler models when used for this same purpose.

Future work will be focused on the independent enhancement of different parts of the system. For example, for the case base, other topology preserving models can be used in combination with the ensemble meta-algorithm to improve the organization of the different cases. Other techniques in the field of CBRs and multi-agent systems will also be integrated into the system to try to improve the currently obtained results.

Generalization is also a future objective of this model, which aim is to be applied to different knowledge fields even if tuning modifications are needed to adapt the model to the new circumstances.

Acknowledgements

This research has been partially supported through project BU006A08 and SA071A08 by the Junta de Castilla y León. The authors would also like to thank the manufacturer of components for vehicle interiors, Grupo Antolin Ingeniería, S.A. in the framework of the project MAGNO 2008 – 1028 – CENIT Project funded by the Spanish Ministry of Science and Innovation.

References

[1] Corchado, J.M., Bajo, J., De Paz, Y. and Tapia, D.I. (2008) Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care, *Decision Support Systems*, In Press.

[2] Yang, J. and Luo, Z. (2007) Coalition formation mechanism in multi-agent systems based on genetic algorithms, *Applied Soft Computing Journal*, 7 (2), 561-568.

[3] Jayaputera, G.T., Zaslavsky, A.B. and Loke, S.W. (2007) Enabling run-time composition and support for heterogeneous pervasive multi-agent systems, *Journal of Systems and Software*, 80 (12), 2039-2062.

[4] Bratman, M.E., Israel, D. and Pollack, M.E. (1988) Plans and resource-bounded practical reasoning, *Computational Intelligence*, 4, 349-355.

[5] Aamodt, A. and Plaza, E. (1994) Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, *AI Communications*, 7 (1), 39-59.

[6] Corchado, J.M., Lees, B. and Aiken, J. (2001) Hybrid instance-based system for predicting ocean temperatures, *International Journal of Computational Intelligence and Applications* 1(1), 35-52.

[7] Carrascosa, C., Bajo, J., Julian, V., Corchado, J.M., *et al.* (2007) Hybrid multi-agent architecture as a real-time problem-solving model, *Expert Systems With Applications*, 34 (1), 2-17.
[8] Fdez-Riverola, F., Iglesias, E.L., Díaz, F., Méndez, J.R., *et al.* (2007) Applying lazy learning algorithms to tackle concept drift in spam filtering, *Expert Systems With Applications*, 33 (1), 36-48.
[9] Karayiannis, N.B. and Mi, G.W. (1997) Growing radial basis neural networks: merging supervised andunsupervised learning with network growth techniques, *Neural Networks, IEEE Transactions on*, 8 (6), 1492-1506.
[10] Cerami, E. (2002) Web Services Essentials Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly & Associates, Inc.
[11] Baruque, B; Corchado, E; Rovira, J; and González, J (2008) Application of Topology Preserving Ensembles for Sensory Assessment in the Food Industry. In Intelligent Data Engineering and Automated Learning - IDEAL 2008: 491-497
[12] Corchado, E., Baruque, B., Yin, H.: Boosting Unsupervised Competitive Learning Ensembles. (ICANN 2007), 4668 (2007) 339-348
[13] Pölzlbauer, G. (2004) Survey and Comparison of Quality Measures for Self-Organizing Maps. In Rauber, J.P.a.G.P.a.A., Fifth Workshop on Data Analysis (WDA'04). Elfa Academic Press, 67--82.
[14] Kohonen, T.: The Self-Organizing Map. Neurocomputing, 21 (1998) 1-6
[15] Ros, F., Pintore, M. and Chrétien, J.R. (2007) Automatic design of growing radial basis function neural networks based on neighboorhood concepts, *Chemometrics and Intelligent Laboratory Systems*, 87 (2), 231-240.
[16] Plaza, E., Armengol, E. and Ontañón, S. (2005) The Explanatory Power of Symbolic Similarity in Case-Based Reasoning, *Artificial Intelligence Review*, 24 (2), 145-161.