

A Multi-Agent System for the Solution of Fuzzy Incidents in Telecommunications

Lilyam Paolino, Horacio Paggi, Genoveva López*, Fernando Alonso Amo*

Universidad ORT Uruguay,

Facultad de Ingeniería, Montevideo, Uruguay

{lilyam.paolino, horacio.paggi}@gmail.com

*Universidad Politécnica de Madrid, Facultad de Ingeniería, Madrid, España

{[glopez](mailto:glopez@fi.upm.es), [falonso](mailto:falonso@fi.upm.es)}@fi.upm.es

Abstract.-This article presents a multi-agent expert system (SMAF), that allows the input of incidents which occur in different elements of the telecommunications area. SMAF interacts with experts and general users, and each agent with all the agents' community, recording the incidents and their solutions in a knowledge base, without the analysis of their causes. The incidents are expressed using keywords taken from natural language (originally Spanish) and their main concepts are recorded with their severities as the users express them. Then, there is a search of the best solution for each incident, being helped by a human operator using a distance notions between them.

Keywords-incident, severities, fuzzy incident, multi-agent expert system, distance.

I. INTRODUCTION

Many different initiatives have been undertaken to use expert systems in order to suggest solutions to incidents. Some of them are mathematical models [1], [2], [3] while others are used just as prototypes [4]. In the communications field there are commercial tools which suggest solutions to the users, allowing them to participate in the incident formulation, but excluding them from the search and the later learning of solutions phases [5], [6]. Some projects are focused in other areas such as the education [7] or the knowledge management [8]; there are also network management models providing automation in the maintenance process, regardless of the users perceptions and needs in the detection of incidents severities[9].

SMAF was analyzed, designed, implemented and tested as a multi-agent system, which allows the users the input of incidents with their severities and show them the solutions, which may or may not be confirmed by the user. In SMAF all the agents are intelligent, proactive and semi-autonomous, they interact between themselves and with humans who can be local or mobile in a computational network. SMAF was tested in the telecommunications area (for incidents such as "low speed in the network", "communication problems between hosts", "jammed network", etc.), but it could be

also applied to suggest solutions in other fields such as, electronic commerce, decision making, medicine, etc., just by changing the knowledge base.

Article layout: in II the description of works which take place in this area. In Section III terminology used. In Section IV the algorithm which searches for the solutions. In section V an example of the working algorithm, in Section VI the language used to model the system, in VII the agents' structure and its functioning, in VIII the technology for its implementation, in IX Results obtained during testing process and in Section X Conclusions and possible future extensions.

II. RELATED WORK

Some multi-agent systems work in dynamic environments finding distributed solutions and using certain degree of autonomy [10]. Other systems design models for the agent coordination in a multi-agent system, based on fuzzy clock. In this case it integrates a concurrent engineering model in which each fuzzy clock measures the time elapsed between tasks executions[11]. The aim of other systems is to show the negotiation between agents, in order to solve previously assigned problems where the fuzzy logic intervenes in the determination of task priorities[12], [13].

The authors are not aware of any other system with the ability to register incidents with their severities, allowing the capture of the language's mistakes and its perceptions, finding their solutions as an extension of the human mind through a multi-agent system and interacting with the user, who will confirm or reject the solution found.

III. TERMINOLOGY

Subject. It is any hardware or software in which an incident may happen.

Severity. It is a linguistic variable related to the subject which indicates the degree of incident occurrence.

Incidents. They are events which may cause interruptions or reductions in the service quality, as they are defined in

other knowledge areas such as the Information Technology Infrastructure Library (ITIL) [14], Control Objectives for Information related Technology (COBIT) [15]. In SMAF, it is the combination of the event of a subject with its respective severities and solutions (if it has them). They may or may not be compound by other incidents, called sub-incidents, which can be classified as ‘obligatory-binary’, ‘obligatory-non binary’, ‘non obligatory-binary’, ‘non obligatory-non binary’. A child incident is ‘Obligatory’ when all the possible parent incidents must happen in order for it to happen. It is ‘non obligatory’ when its occurrence does not determine the parents occurrence. The binary can only admit two values, i.e: ‘the network works’ or ‘the network does not work’. Each incident may have 0, 1 or many parents (Fig.1).

Fuzzy incident: It is an incident that can occur in a variable degree or with an imprecise intensity or severity[16], (e.g. “poor speed when browsing the web”).

IV. ALGORITHM WHICH SEARCHES FOR THE SOLUTIONS

The users express the fuzzy incidents (from now on they will be called ‘originary incidents’), with its imprecise severities and input ascending according to their impact to one subject. If no solutions were found to the originary incident, SMAF will search for candidate incidents which could contain the solution to the originary incident. For this it is the following logical sequence:

A.. Severity Sub-Interval Calculation

The linguistic variable values are distributed in subintervals corresponding to the of different severities; each one of them is calculated using the formula :

$$\text{Sub-interval} = [i*(10/\text{total}), (i+1)*(10/\text{total})] \quad (1)$$

where i is the position beginning at 0, and t is the total of sub-intervals registered. E.g. if “the network” (A) is the subject of the incident and the problem is “the slowness”, the intervals are set in ascending order according to the impact, it would be ‘it works slowly’ [A(0, 2.5)], “it works very slowly” [A(2.5, 5)], “it works too slowly” [A(5, 7.5)] “it does not work” [A(7.5, 10)]. The union of the severities intervals of a subject is 10, and its intersection is an empty set.

B. Formation of Sets by Contentment

For each of the originary incident, the incidents whose intervals contain the first are searched (solution by contentment) and then a new set is form with all of them. E.g. for the A(3, 7) incident, the plausible results by contentment are the incidents on A with the intervals containing the severities from 3 to 7, as 0 to 7, 2 to 7, 3 to 8, etc.

C. Formation of Sets by Proximity

If there is not any candidate incident within the contentment, the best candidates are going to be those which are closer to the originary interval. E.g. if the originary incident is A (4, 6) and in the data base are A (0, 2) and A(9, 10), the incident within the closest severity range would be A (0, 2). As there is a difference of 2 between the originary incident and the candidate: (4 - 2 = 2). Meanwhile, the difference with the other incident set is 3 , because (9 - 6 = 3).

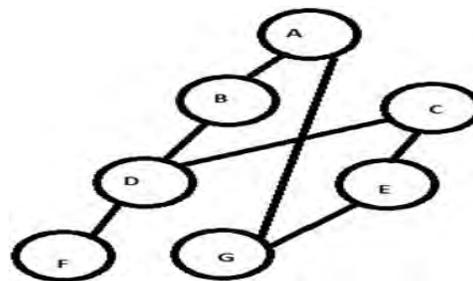


Figure 1. SMAF distance example.

D. New Set Formation by Distance Evaluation.

A new set is formed, integrated by the sets obtained in B and C and all their ascending and descending.

The distance is the minimum quantity of nodes which separate the nodes among themselves, being 0 the distance to itself, 1 the distance to its parent, and so on. If in one particular set there is an incident repetition, the only distance to be considered will be the shortest.

In fig. 1, with the supposition that the originary incidents are F and G and applying the algorithm for the search (IV), the incidents which could contain the solutions to F and G are A and C. The best solution to the incidents is the one with the shortest distance to the originary incident. The total distance (TD) is the sum of the distances between candidates nodes and the originary nodes in a square base. By this way, the higher distances are penalized.

In the case of figure 1, the distance between A to F is 3 and the distance between A to G is 1, therefore, the TD from A is $3^2 + 1^1 = 10$. From C to F is 2 and from C to G is 2, so the TD from C is $2^2 + 2^2 = 8$. So the solution of C is better than the solution of A, because C is closer than A to the originary incidents

E. Formation of the Result Set.

To find all the incidents which contemplate all the problems, there has to be done an intersection of the sets obtained in D and they are presented in ascending order, according to their distances, in order to present the candidate solutions to the user. The user will later confirm or not the solution to the incident. If the user rejects the solution, the incident will be transferred to an expert for its treatment.

V. AN EXAMPLE OF THE OPERATION OF THE ALGORITHM

To perform the tests of the presented algorithms, subjects with different severities were input (A, B, C, D, E, F, G, H, I, J). SMAF encoded their severities (Table 1) by (1) and stores all the data in a knowledge base and shows a graph (fig. 2) with the relationships among incidents. Then, for example, if the user is looking for a solution to the C (0, 2) and E (0, 2), it finds that the C (0, 4) contains the wanted C (0,2) and E (0, 3) contains E(0.2). Considering C (0, 2) and C (0, 4) incidents, SMAF gets a new set of incidents called AA (table 2), which has all its children and ancestors. So for the E (0, 2) and E (0, 3), SMAF gets another set of incidents named BB (Table 3). AA and BB are candidate sets that have all the incidents whose solutions could solve C(0,2), C(0,4), E(0,2), E(0,3), (Solved by closeness) .Then the SMAF makes an intersection between AA and BB (Table 4) and finally the solutions are presented by the user from bottom to top, depending on the distances(Table 5).

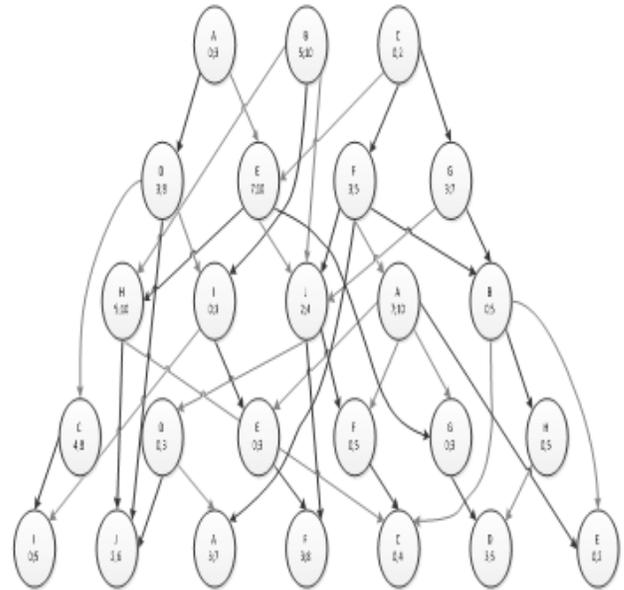


Figure 2. Incidents graph

TABLE I. DATA INPUT (SUBJECT, SEVERITIES RANGES)

Subj.	Severity ranges	Subj.	Severity ranges
A	0;3 – 3;7 – 7;10	F	0;3 – 3;5 – 5;8 – 8;10
B	0;5 – 5;10	G	0;3 – 3;7 – 7;10
C	0;2 – 2;4 – 4;6 – 6;8 – 8;10	H	0;5 – 5;10
D	0;3 – 3;5 – 5;8 – 8;10	I	0;2 – 2;3 – 3;5 – 5;7 – 7;8 – 8;10
E	0;2 – 2;3 – 3;5 – 5;7 – 7;8 – 8;10	J	0;2 – 2;4 – 4;6 – 6;8 – 8;10

TABLE II. AA SET

Incident	D	Father av
C(0,2)	0	
C(0,4)	0	
H(5,10)	1	C(0,4)
B(5,10)	2	H(5,10)
E(7,10)	2	H(5,10)
A(0,3)	3	E(7,10)
C(0,2)	3	E(7,10)
B(0,5)	1	C(0,4)
F(3,5)	2	B(0,5)
G(3,7)	2	B(0,5)
F(0,5)	1	C(0,4)
A(2,4)	2	F(0,5)
A(7,10)	2	F(0,5)

TABLE III. BB SET

Incident	D	Father Av
E(0,2)	0	
E(0,3)	0	
I(0,3)	1	E(0,3)
A(7,10)	1	E(0,3)
D(3,8)	2	I(0,3)
B(5,10)	2	I(0,3)
A(0,3)	3	D(3,8)
F(3,5)	2	A(7,10)
C(0,2)	3	F(3,5)
B(0,5)	1	E(0,2)
G(3,7)	2	B(0,5)

“D” means Distance.

TABLE IV. INTERSECTION AA-BB

Incident	D
C(0,2)	9 (0 ² + 3 ²)
B(5,10)	8 (2 ² + 2 ²)
A(0,3)	18 (3 ² + 3 ²)
C(0,2)	18 (3 ² + 3 ²)
B(0,5)	2 (1 ² + 1 ²)
F(3,5)	8 (2 ² + 2 ²)
G(3,7)	8 (2 ² + 2 ²)
A(7,10)	5 (2 ² + 1 ²)

TABLE V. SORT INTERSECTION AA-BB

Incident	D
B(0,5)	2
A(7,10)	5
B(5,10)	8
F(3,5)	8
G(3,7)	8
C(0,2)	9
A(0,3)	18
C(0,2)	18

VI. LANGUAGE USED TO DESIGN SMAF

The standard language used when designing the system was AUML. This allowed the representation of the agents, their social structures and organization presented in a deployment diagram. It also allowed the representation of the components of the system's functionalities with its expert agents, their respective frames, the agents' structures and the sequences to release the incidents' subjects and to the search algorithm (Fig. 3).

VII. AGENTS' STRUCTURE AND IT'S FUNCTIONING

SMAF is formed by heterogeneous agents in structure and functions. They have specific responsibilities which make them expert in their areas, they also can interact with other agents, whether they are people or software agents, through messages and keeping their semi-autonomy. As they sometimes begin their actions because of the agents (humans or software) and other times they begin their actions by themselves.

Each agents has incomplete information to solve the specific tasks, so they can make decisions based on their knowledge and based on the information transferred by other agents.

The agents may be local, from the same computer, or mobile, if they move within the computational network through the recognition of the IP addresses and the hosts ports and keeping the recursive references of the data in the knowledge base that they need to accomplish their tasks. If they can recognize their own status, they will know whether they are active or latent, as that information is integrated in their structure as dynamic data. They may also have a recursive structure and be compound by other agents. e.g. Agent Incident which is formed by lists that have the ascending and descending agents, and also the specification of the different types of children: obligatory, non-obligatory, binary and non-binary. There is also certain degree of pro-activity as they can receive input of their environment (e.g. a new subject or incident) and do their actions (e.g. adjust the incidents' data base), as a result of their decisions (e.g. define which is the best solution). They are coherent, because they record their solutions in a knowledge base, so they learn how to recognize the best solutions in further searches.

The **'SubjectAgent'** validates, releases, modifies or deletes the subject the users input, interact with the data Access Agent. It also does the interface with the user but it can act autonomously. E.g. detecting an incident which does not belong to any subject, requesting its entrance or erasing it, or modifying it, if the associated incident was released or modified.

'IncidentAgent'. It validates, modifies or deletes the users input regarding the incident. In its structure it has the event itself, with its severities and its possible solutions (if there were any) and the incident's subject references to its parents and children. It interacts with the DataAccessAgent. It acts as the users ask for it, but it also can act automatically if it detects any incongruence in the data base, and manages the congruence in the graph. E.g. if some of the children or the parents are deleted, they can make their own decisions

'SearchAgent'. It interacts with the 'input output user agent'. It receives the incidents and searches in the graph the incidents that represent the users' problems. It has a matrix the subjects, the incident, their severities and their solutions (if there were any). It acts if the users requests it, but it can act autonomously if it finds a better solution in the data base. E.g. a shorter distance than the previously found, using its intelligence.

'InputAndOutputAdministratorAgent' (GUIAdministratorAgent) and **'InputAndOutputUserAgent'** (GUIUserAgent). They are interface agents. They send the validated subjects and incidents to the Data Access Agent and it interacts with the administrator and the user, respectively.

VIII. TECHNOLOGY

JADE was used because of the facilities it gives to the agents interaction through messages, being also a useful tool for the data and knowledge distribution. The agents' communication is done through a method invocation where an ACLMessage object is send, and according to the object it sends a message. E.g. When the GUIAgent receives the new incident's data to be released, it sends a message to the IncidentAgent through the AltaIncident which gathers all the data received, and it is the responsible to validate the data and to allow the new incident. (Fig. 3). SMAF was develop for Windows.

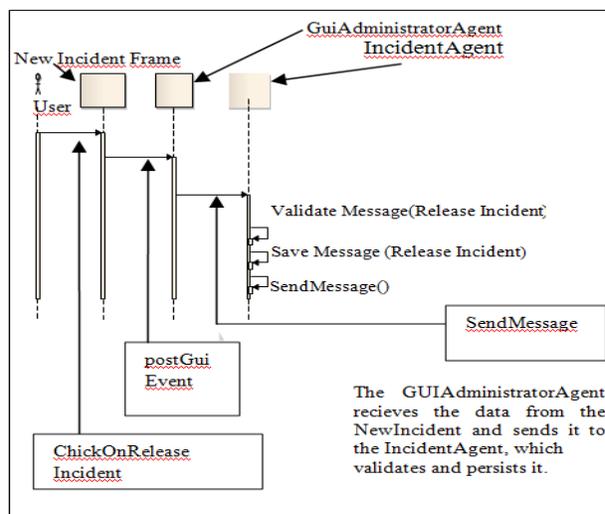


Figure 3. One of the sequence diagram (Incident release)

IX. PRELIMINARY RESULTS

Of a total of 64 incidents input with their respective severities in order to find their solutions, 58 of the found solution were considered correct by the users, meaning a 90,6% of correct answers to the requested solutions.

X. CONCLUSIONS AND FUTURE EXTENSIONS

SMAF, a multi-agent system accomplished its objective, which was to release the incidents with their different severities and to find better solutions. To do so, an algorithm to release the incidents and subjects was defined, and another algorithm to the search of solutions. It was designed through AUML and implemented with JADE tool. SMAF was tested in the telecommunications area.

It seeks a solution for the incidents in the input incident itself. If it does not find one, the system searches in other incidents, within a given range (Solved by contentment). If through this process it does not find a solution, it searches for an incident whose range is close enough to the original. The solution found is presented to the user following certain priority order, based on the distances between the original incident and the candidate incident found. If the user is not satisfied with the solutions, SMAF transfers the original incident to an expert.

One possible improvement could be to define the maximum severity range as a variable (i.e. the ranges of severity variable are not just [0, 10]). Another improvement could be to perform different treatments, depending on the kinds of children, prioritizing the searches for 'obligatory children'. Another plausible improvement could be to evaluate the system's functions, bearing in mind the amount of correct answers per user in relation with the number of requests, and to seek a solution for the incidents. Other enhance to SMAF could be the integration of other servers that could have other possible solutions to the incidents.

SMAF was developed in a computational intelligence area to try to improve the knowledge management of solutions found for fuzzy incidents. It was tested in the telecommunications area but it could be also applied to suggest solutions in other fields.

ACKNOWLEDGMENT

Thanks to Andrés Ferraro, Matías Comesaña, and Christian Sardas from ORT University in Uruguay, for their valuable inputs in the investigative process of system SMAF.

REFERENCES

- [1] K. Atanassov "Intuitionistic fuzzy sets : past, present and future" in "Fuzzy Set and System", vol 31, Nro. 3, pag 343-344, 1989.
- [2] D.Wu and W. W. Tan " Design of type-reduction strategies for fuzzy 2. Fuzzy logic systems using genetic algorithms" in Advances in Evolutionary Computing for System Design, Eds.L.C. Jain, V. Palade and D. Srinivasan, Springer 2007, pp. 169-188.
- [3] J.M. Mendel and D. Wu "Computing with words for hierarchical and distributed decision making", Ch. 9 in Computational Intelligence Complex Decision Systems, D.Ruan, Ed. Paris, France : Atlantis Press, 2009..
- [4] L. Amable García Fernández, "Design and implementation of a multi-agent architecture to make decisions in a urban traffic system ", <http://www.tesisenxarxa.net/TDX-0528108-145852/> , <http://www.Kriptia.com/list/es/universidad/Valencia/> , 1999 .
- [5] Xperta.<http://www.asac@asac.as/> (Xperta : System about incidents management, developed by Asac, 1996)
- [6] <http://kelvintopset.com> (Kelvin : Tool about incidents management developed by Top Set, 1999).
- [7] J Soto , E García Gordo "Intelligent multiagent system for the study of a student organized planning", "Sistema multiagente inteligente para la planificación organizada del estudio de un alumno", Universidad Pontificia de Salamanca, Departamento de Inteligencia Artificial, Lenguajes, Lenguajes formales y Matemática Aplicada, Madrid, España, 28040, jesus.soto@upsam.net, elisa.garcia@upsam.net
- [8] Cobos, R, Pifarré, M. "Collaborative Knowledge Construction in the Web supported by the KnowCat system", in Computers & Education 50, pp. 962-978. Elsevier Ltd, United Kingdom. ISSN: 0360-1315, 2008
- [9] A. Maj, J. Jurowicz, J. Kořlak, K. Cetnarowicz, "A multiagent System for Dynamic Network", V. Mařik et al. (Eds): CEEMAS 2003, LNAI 2691, pp. 511-521, 2003. Springer-Verlag Berlin Heidelberg 2003
- [10] A. Niimi, O. Konishi, "Data Mining for Distributed Databases with Multiagents", "Proc. Knowledge-Based Intelligent Information and Engineering Systems – KES., Pp. 1412-1418, 2003
- [11] J.Peters, N.Sohn "Coordination of Multiagent Systems with Fuzzy Clocks" Department of Electrical and Computer Engineering. University of Manitoba, ENGR 504, Winnipeg, Manitoba R3T 2N2 Canada. March 1996 4: 73-87.
- [12] D.Wu and W. W. Tan "Design of type-reduction strategies for type-2 fuzzy logic systems using genetic algorithms," in Advances in Evolutionary Computing for System Design, Eds.L.C. Jain, V. Palade and D. Srinivasan, Springer 2007, pp. 169-188.
- [13] L.K.Sohn, C.Tsatsoulis "Reflective Negotiating Agents for Real Time MultiSensor Target Tracking", in Proceedings of IJCAI, 2001, Seattle, WA, 1121-1127.
- [14] ITIL–United Kingdom's Office of Government Commerce (OGC) [http:// www.opengeospatial.org](http://www.opengeospatial.org)
- [15] COBIT-Control Objectives for Information and related Technology <http://isaca.org>
- [16] J.M. Mendel and D.Wu , "Computing with words for hierarchical and distributed decision making ", Ch. 9 in Computational Intelligence in Complex Decision Systems, D.Ruan, Ed. Paris, France : Atlantis Press, 2009.