Aspects Related to Methodological Design of Multi-agents Systems in Anesthesiology

B. Drozdowicz[+*], A. Hadad[*], D. Evin[*], C. Böhm[^] and O. Chiotti[+^]

+ Instituto de Desarrollo y Diseño (INGAR) – CONICET – SANTA FE – Argentina
* GIAA - Facultad de Ingeniería – Bioingeniería – Universidad Nacional de Entre Ríos
^ GIDSATD-UTN – Facultad Regional Santa Fe –Argentina
bdrozdo@ceride.gov.ar

Abstract. In this paper aspects related with the methodological design of a multi-agent system to support both supervision of anesthesiological processes and anesthesiologists decisions are approached. This work describes how it was arrived to a multi-agent architectural design complementing the Gaia methodology with an approach for roles identification. Then, it shows the ActionPlanSupervisor Role development using the Situation Calculus as a basis to model the domain evolution by actions (strategies performed by anesthesiologists). In this model the use of fuzzy relationships to represent preconditions, actions and post-conditions are proposed. The advantage of this representation is based on reaching a better interpretation and more flexibility as regards to the characteristics of each patient and the interaction with the anesthesiologist. In this way, this representation adds clearness to the domain analysis and a familiar language to the application field.

1 Introduction

The design of an intelligent system for the supervision and decision support of complex domains generally requires tools and methodologies to make its analysis and development easier, bringing about the necessary flexibility, adaptability and modularity in order to obtain an efficient and adaptable design. The complex domain considered in this paper is related to the anesthesiology processes that are necessary to implement in every surgery. Anesthesia is a procedure to suppress patients' abilities to feel, to move, and/or to remember. It is used in a wide range of medical procedures, mostly to provide suitable patient conditions for surgery. In the majority of surgical cases, anesthesiologists induce patients into an unconscious state prior to the beginning of the surgery, maintain that state during the course of the surgery, and then terminate the process at the end. During anesthesia, it is primarily the anesthesiologist's responsibility to ensure the patient's well-being and to compensate for the effects of surgery and of the anesthesia itself, be they intentional or accidental.

In this contexts the professional's human error probability is increased because of the unfavorable conditions: high level of stress, lack of time, variability of situations, interaction among different professional groups, etc. Thus, an information system to

improve the information quality could be important, in order to increase the patient safety and the anesthesiologists quality of work [1].

The decision process involved in a complex domain can be divided in several sub-processes, where each of them will carry out autonomous actions based on appropriate domain knowledge. To carry out its actions, a sub-process may require some decisions of other sub-processes and the result of it decisions may be required by other sub-processes. These characteristics of a complex domain allow inferring that the agent-based software technology can be appropriate to develop an intelligent system to support such a complex domain.

A survey of agent-oriented software engineering presented in [2] describes recent advances of methodologies to develop agent-based systems. Analyzing the characteristics of the anesthesiology problem domain, it is concluded that they match the domain requirements of Gaia methodology [3]: closed-limited work place, reduced set of agents with static services and abilities inter-agents relationships.

Therefore, Gaia methodology was chosen to develop a multi-agent architecture of a system in order to support the anesthesiology domain. This methodology proposes a developer to think of building agent-based systems as a process of organizational design. The main concepts in Gaia are divided into two categories: abstract and concrete. The abstract concepts involve the *Roles*. A role is defined by four attributes: *Protocols, Permissions, Activities* and *Responsibilities*. Responsibilities are divided into two types: *Liveness Properties* and *Safety Properties*. The concrete concepts involve: *Agent types, Services* and *Acquaintances*. A role can be viewed as an abstract description of an entity's expected function. In other words, a role is more or less identical to the notion of an office. A protocol defines the way that a role can interact with another role. The analysis stage implies defining the role model and the interaction (protocols) model. In this work, the terms of Gaia are defined in the different sections where they are necessary.

Although Gaia claims to allow an analyst to go systematically from a statement of requirements to a sufficiently detailed design, it does not present a procedure to guide roles and protocols identification. Thus, it was decided to apply the approach described in [4] which allows the roles and protocols identification viewing the system as set of processes. Furthermore, an extension to this approach is proposed, using an activity diagram to represent the flow control of the tasks set required to carry out each activity of the defined processes. This approach allows for an easier definition of the liveness property of a role.

In order to make the proposed methodologies for agent-based systems development become a new paradigm for the software industry, robust and easy-to-use methodologies and tools have to be developed [2]. That is, this is not a mature research area. That is why the object of this paper is to present a strategy to develop an agent-based system architecture for a complex domain with the purpose of contributing to consolidate this research field. Firstly, the development of the multi-agent system architecture is presented. Secondly, the main role of this domain, the ACTIONPLANSUPERVISOR role, is described.

2 Development of the Multi-agent Architecture

The approach described in [4] allows the roles and protocols identification to view the system as a set of processes. This approach consists of 3 stages:

- *Stage 1:* To define the system's goal.
- *Stage 2:* To define processes through their inputs, outputs and activities.
- *Stage 3:* To identify the necessary roles and protocols in order to perform the previously mentioned activities.

Once the third stage is completed, the Gaia methodology may be applied.

2.1 Stage 1: To Define the System's Goal.

For this case of study the system´s goal is defined as follows: "To reduce the possibility of patients harm during the anesthesia process, giving support to the anesthesiologist in keeping such a process under normal conditions."

2.2 Stage 2: To Define the Process and its Activities

The system is modeled as a process, defining the inputs, outputs and precedence order of the activities that conform it. The system consists of just one process: "Giving support to the anesthesiologic process". It is represented in Figure 1.

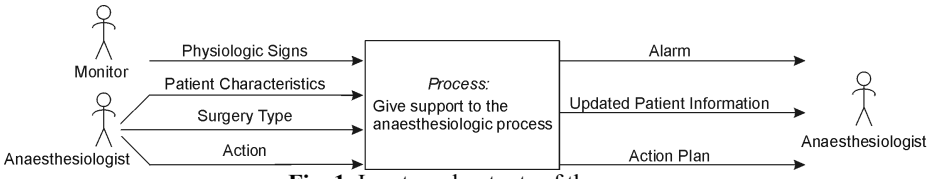


Fig. 1. Inputs and outputs of the process

The inputs for this process are:

- *Physiologic Signs:* they are periodic samples of the patient vital signs (Heart rate, blood pressure, etc.) obtained from the existing monitors. They are used to infer the patient state.
- *Patient characteristics:* descriptive patient information. It comprise both static data (age, sex, weight, etc) and periodic data (clinical lab test). They are loaded by the anesthesiologist before the surgery.
- *Surgery Type:* It indicates the specific surgery type. The anesthesiologist selects them from a previously loaded surgery-types list.
- *Action:* It indicates to the system the action that has been done by the anesthesiologist. It allows updating or redefining the action plan proposed by the system.

The system outputs help the anesthesiologist to improve their activities:

- *Alarms:* the standard equipments monitoring the patient produces a high level of unnecessary or false alarms. They are produced by events not related to the actual patient state. They are the most frequent nuisance and confusion sources for the

surgery staff in charge. Therefore, the system will activate them only in necessary cases.

- *Updated patient information and action plan:* the system will give the anesthesiologist updated information of the patient and will suggest an action plan to follow during normal and abnormal conditions (emergency).

Figure 2 shows the activities diagram of the process.

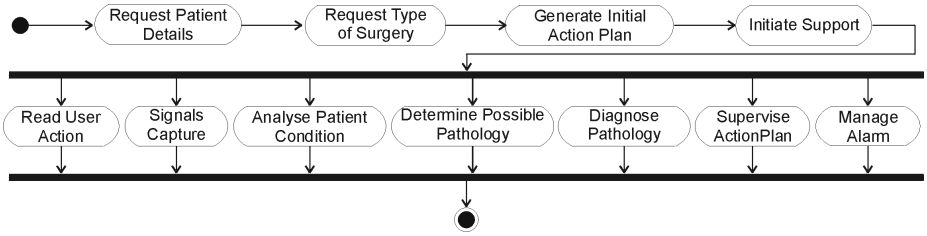


Fig. 2. Process activities diagram

The process is divided in two phases: configuration phase and support phase. The configuration phase consists of the first four process activities, while the support phase consists of the set of activities it will execute concurrently until the process finish.

During the configuration phase, the first two activities allow the anesthesiologist to load the system with the patient characteristics and with the surgery type. Finally, the last two activities correspond to the system initialization and preparation for the next support phase.

As it was previously mentioned, during the support phase the activities will be executed concurrently. These activities will effectively give support to the anesthesiologist during the surgery.

The diagram can not model the fact that each activity within the synchronization bars will be executed iteratively until a signal indicating the process finalization is received. This aspect would be shown when these activities are modeled more in detail.

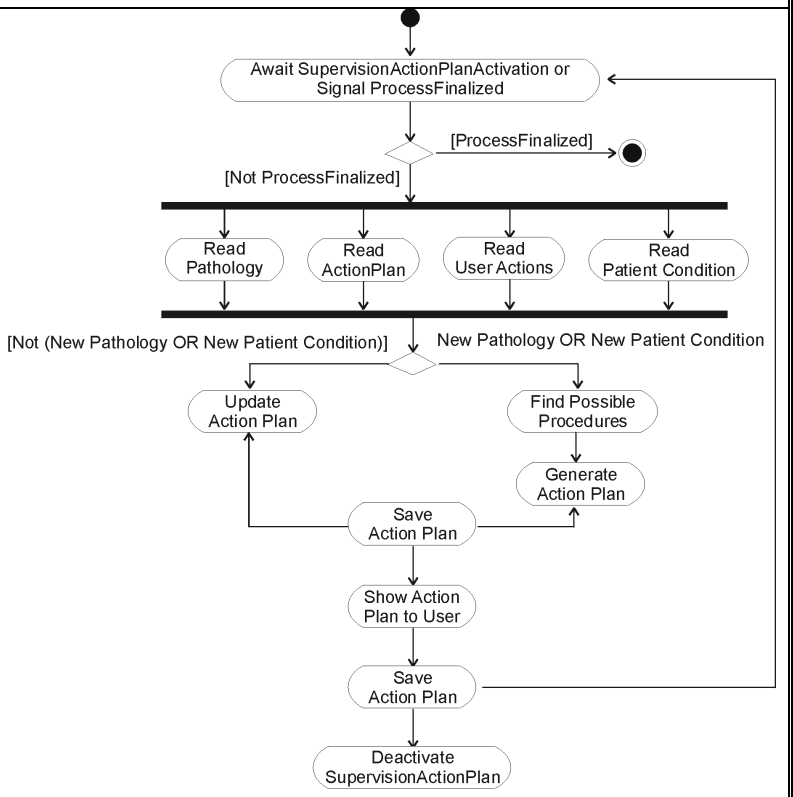
To complete the second stage, for each process activity a template describing their different attributes: objectives, inputs, outputs, tasks list, restrictions and resources, was made.

Due to the complexity of the control flow in several process activities, this work proposes to make an activity diagram instead of a tasks list as originally proposed in [4]. Such a diagram not only allows the loops, alternative flows and concurrent tasks modeling, but it also helps to define the liveness properties as well.

As an example, the activity template Supervise Action Plan is shown in Table 1 and, later in this paper, it would be implemented as an agent role.

Table 1 shows the concurrent activity tasks Read Pathology, Read Action Plan, Read User Actions and Read Patient Condition and the fact that the activity task Update Action Plan can be executed alternatively with the activity tasks Find Possible Procedures and Generate Action Plan. This fact would not have been possible to represent if a simple tasks list had been used as proposed in [4].

Table 1. Supervise Action Plan Activity Termplate

Activity	Supervise Action Plan
Object	Using the patient condition, user actions and diagnosed pathologies, generates and updates the action plan to suggest.
Inputs	Pathology, User Actions, Action Plan, Patient Condition, Procedures
Outputs	Action Plan, Signal Deactivation SupervisionActionPlan
Tasks	
Constrains	-
Resources	Reads: Pathology Base, User Action Base, Action Plan, Patient Condition Base. Writes: Action Plan.

2.3 Stage 3: To identify Roles and Protocols

The activities carried out during the previous stages allow a simple implementation of this stage since the necessary information to describe roles and protocols according to the Gaia methodology has been already collected. The next step would consist of grouping related activities into roles.

The role attributes proposed in Gaia are: Description, Protocols and Activities, Permissions and Responsibilities. Following the Supervise Action Plan activity example, Table 2 shows its implementation as a role.

Table 2. ActionPlanSupervisor Role

Role: ActionPlanSuperVisor
Description: Realizes the Supervise Action Plan activity. This role will be active until it receives the ProcessFinalized signal.
Protocols and Activities: AwaitSupervisionActionPlan, ReadPathology, ReadActionPlan, ReadUserAction, ReadPatientCondition, UpdateActionPlan, FindPossibleProcedures, GenerateActionPlan, SaveActionPlan, ShowActionPlanToUser,DeactivateSupervisionActionPlan.
Permissions: Reads Signal ActivationSupervisorActionPlan Signal DeactivationSupervisorActionPlan Signal FinalizeSupervisor Base Pathology Base UserActions Base PatientCondition ActionPlan KB Procedures Generates ActionPlan Signal DeactivationSupervisorActionPlan
Responsibilities **Liveness:** SupervisorPlanAccion= (AwaitSupervisionActionPlan.[(ReadPathology || ReadUserAction || ReadActionPlan || ReadPatientCondition).(UpdateActionPlan | (FindPossibleProcedures. GenerateActionPlan)). SaveActionPlan, ShowActionPlanToUser. DeactivateSupervisionActionPlan])*

Note that due to the role complexity, it would have been very difficult to write the expression liveness property if the activity diagram proposed in the SuperviseActionPlan Activity Template in Stage 2 had not been performed before.

Once Stage 3 is finished it is possible to apply the Gaia methodology in order to obtain the Agent, Interaction, Acquaintance and Services Model. As there is not space enough to fully explain all these models, only the Agent and Acquaintance models are presented in this work. The former is shown in Figure 3 and the latter is shown in the multi-agent architecture in Figure 4.

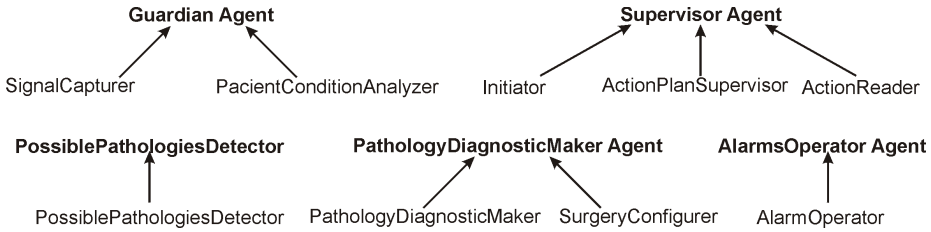


Fig. 3. Agent Model

As it can be seen, some agents are mapped to more than one role. For example, the Supervisor Agent is in charge of supervising the action plan, generating the initial action plan, telling the other agents that the supervision is starting and also, reading the actions carried out by the anesthesiologist.

2.4 Multi-agent Architecture

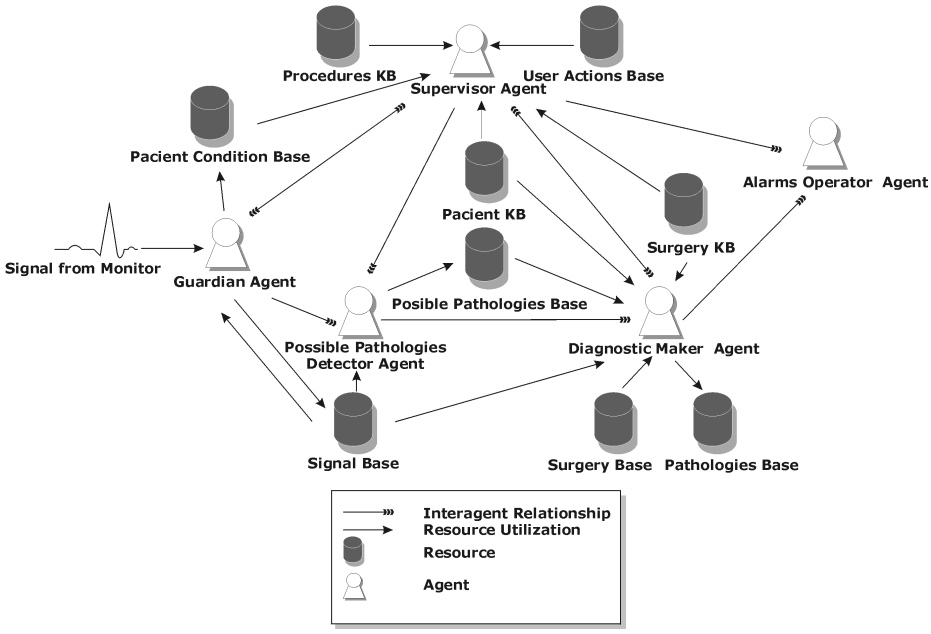


Fig. 4. Multi-agent architecture

Using the information gathered through the previous stages, the multi-agent architecture shown in Figure 4 was designed. This architecture shows the resources that each agent utilizes and the relationships among them.

The following section describes the ACTIONPLANSUPERVISOR role defined in previous stages.

3 Description of the ActionPlanSupervisor Role

This role plans the strategies, taking into account the current patient state and different actions that can be made by the user during the anesthesiologic process. For such planning, a model based Situation Calculus is used. It allows to represent the world changes caused by actions (i.e. activities performed by the anesthesiologist). In the model, the world characterization (i.e. the patient state with respect to the anesthesiologic process) is performed by functional and relational fluents, which involves the most important information [5].

3.1 Strategies Planning

Firstly a final goal must be defined, which in this case will be to maintain the patient in normal condition, reversing all possible abnormal situations that could happen during the anesthesiologic process.

A patient characterization can be obtained using his biological signals, so that the fluents would take specific values. Such values, complemented with other agents information, satisfy the preconditions of a finite number (n) of actions (a_n). Then those actions are used separately to simulate their effects in state S (in this case the initial state $S = S_0$); as result of the different simulations, n states (S_1', S_2', ..., S_n') will be obtained. From these n states, the previous process can be repeated until the final goal is satisfied. As a result of this process, one or several actions combinations could be obtained. Each of these action combinations is called strategy. As a consequence of this analysis, the following situations may arise:

- Only one possible strategy that satisfies the goal is identified by the model. The system informs the anesthesiologist about this simulated strategy as well as its warnings and limitations based on clinical criteria available by the agent.
- More than one strategy that satisfies the goal is identified by the model. Based on criteria such as: actions effectiveness and risk, the agent must compare the obtained strategies. If two strategies with the same effectiveness and risk were found, the agent would analyze other criteria, for example execution time, and number of actions. The chosen strategy will be shown to the anesthesiologist.

Before sensing the world and updating the fluents again, the agent waits during an interval of time (T). This time interval T depends on the estimated execution time of the anesthesiologist's actions and their effects identification. The information of such an interval is stored in the agent knowledge base. If the agent were waiting to sense the world again and it identifies a critical event, the system would interrupt the current strategy and would take another strategy corresponding to the previously identified event.

Based on his experience, the anesthesiologist may execute a different action to the one recommended by the selected strategy. In this case, there are two possibilities:

1. The action performed by the anesthesiologist causes a similar patient state than the predicted by the agent strategy. In this way, the preconditions are maintained valid for the next planned action. The professional may inform the agent of the performed action, updating the agent's knowledge base with the new experience.

2. The action performed by the anesthesiologist causes a patient state that differs from the one predicted by the agent strategy. In this way, the preconditions values are modified for the next planned action. In this case, the agent must request a confirmation of the action performed by the anesthesiologist. If it is confirmed, the agent will have to generate a new strategy from the current new initial state.

3.2 Special treatment of Pre and Post-conditions

To work with complex systems (like patients), anesthesiologist analysis must be based on his/her previously acquired experience and knowledge from the environment. In making their decisions they may use many factors, which could have a range of validity in accordance to the patient characteristics. In addition, in that analysis, some qualitative fuzzy concepts as: high heart rate, very low pressure, normal ventilation, etc., are used for the evaluation of patient's states and the effects of the actions.

This variability and subjectivity have an effect on the definitions of the preconditions (evaluation of the patient current state), of the actions and preconditions (effects of the actions on the patients). So, in this paper, these model characteristics are proposed to be represented by means of fuzzy sets and to be treated by procedures of the fuzzy logic.

To show this proposed representation, the following example is presented.

In an abdominal gynecology surgery, different doses of the anesthetic drug alfentanil must be injected according to the patient state [6].

In order to simplify the example, only two patient parameters are considered: HR and MAP. The range of 20 to 150 [mmHg] of the MAP was fuzzified in five fuzzy sets, the same as for the range 0 - 200 [bpm] for the HR. These parameters represent the precondition of a group of actions.

Table 3. Preconditions Fuzzy Sets

HR - Heart Rate [1 / min.]			
Very_low	0	25	50
Low	25	50	75
Normal	50	75	100
High	75	100	125
Very_high	100	125	200

MAP - Mean Arterial Pressure [mmHg]			
Very_low	20	40	50
Low	40	60	80
Normal	60	80	100
High	80	100	120
Very_high	100	120	150

In order to evaluate the action effects on the patient, the actions and post-conditions must be also fuzzified. In this case, in order to simplify the example, only one parameter variation (ΔMAP) and only one kind of action modify the drug flow (dose). Therefore, these two variables must be also fuzzified (Table 4).

Table 4. Post-condition and Action Fuzzy Sets

ΔMAP [mmHg]			
Small	0	5	10
Medium	5	10	15
Big	10	15	20

Δ Drug Flow [ug/kg/h]			
Small	0	10	20
Medium	10	20	30
Big	20	30	40

Defined the fuzzy sets, a particular case is supposed:
MAP = 105 mmHg ➔ belongs to the fuzzy sets **high (0,75)** and **very_high (0,25)**
HR = 75 [1/min.] ➔ belongs to the fuzzy set **normal (1)**
These conditions verify the preconditions of the following action:
Preconditions: HR normal and MAP high.
Action: increase the dose of alfentanil. Δ drug flow positive medium.
Post-condition: MAP decrease. ΔMAP negative small.

It is possible to define a fuzzy relation between the preconditions and the actions (modify the drug doses). With this relation the necessary action could be identified (i.e. high increase in the drug dose). Through a defuzzification process, the agent is able to obtain the amount of drug which must be increased (for example 20 ug/kg/h).

It is also possible to define a fuzzy relation between the action (drug dose variation) and the post-conditions (pressure variation). With this relation, the type of post-condition could be obtained (a small positive change of pressure). Through a deffuzzification process, the agent is able to identify the new value of the post-condition (small negative ΔMAP) and consequently the new state [7].

In this way, the fuzzy logic allows the transition between different states of the proposed model in Situation Calculus and developing the possible strategies.

4 Conclusion

Gaia, like any other methodology to develop multi-agent systems, is still subject to research and discussion. Therefore, we consider that the description of a use case in the analysis and design of a system using this methodology makes a contribution to the evaluation of its applicability and its strong and weak points. In this work, we realized the advantage of complementing Gaia [4] with a process that guides the roles and protocols identification at the analysis stage.

Particularly, the role modeling through the system processes and their activities description was in our case a useful and simple tool, allowing us to discover details that it would have been difficult to discover applying an ad-hoc model from which Gaia starts.

Since much of the information in the anesthesiologic field is given in a qualitative fashion, the fuzzy approach characteristics are suitable for the professional – system interface. This representation can be reflected in the system model for the fuzzification of preconditions, actions and post-conditions that will be treated by procedures of the fuzzy logic.

The advantage of this representation is based on reaching better interpretation and obtaining more flexibility as regards the patient characteristics and the system-

anesthesiologist interaction, since it adds clearness to the analysis and a familiar language to the application field.

This kind of treatment is needed since it is desirable to take control of the patient state, as soft as possible, within a "normality range". With the normality range, we are indicating a fuzzy zone of values that can take the patient's biological signals during the complete anesthesiologic process (i.e. no one of the parameters must reach critical values, because it can imply irreversible damages for the patient).

References

[1] Krol M, Reich D. L., Development of a Decision Support System to Assist Anesthesiologist in Operating Room. Journal of Medical Systems Vol 24 (3) (2000 jun).

[2] Tveit, A., A Survey of Agent-Oriented Software Engineering, First NTNU CSDSC, (2001).

[3] Wooldridge,M., Nicholas R. Jennings, David Kinny, The GAIA Methodology for Agent-Oriented Analysis and Design, J. of Autonomous Agents and Multi-Agents Systems. Vol 3(3), (2000).

[4] Villareal, P., M. Alesso, S. Rocco, M. R. Galli, O. Chiotti, Approaches for the Analysis and Design of Multi-Agent Systems, Revista Iberoamericana de Inteligencia Artificial. España.

[5] Reiter, R., Knowlege in action. Logical Foundations for Specifyng and Implementing Dynamical Systems (2001).

[6] Mean arterial pressure and heart rate with alfentanil controlled by fuzzy logic. A. Carregal, A. Lorenz, J.A. Taboada, J.L. Barreiro. Spanish Journal "Anestesiología Y Reanimación". Vol 47 – Nº 3 – pp. 108–113, (2000).

[7] Horia-Nicolai Teodorescu, Abraham Kandel, Lakhmi C. Jain, Fuzzy and Neuro-fuzzy Systems in Medicine Methodological Design of Multi-agents Systems in Anesthesiology.doc, CRC Press LLC, (1999).