Risk prediction in surgery using case-based reasoning and agent-based modelization

Bruno Perez[1], Christophe Lang[1], Julien Henriet[1], Laurent Philippe[1], and FrÃ©dÃ©ric Auber[2−3]

[1] FEMTO-ST Institute

Univ. Bourgogne-Franche-ComtÃ©, CNRS

DISC, 16 route de Gray, 25030 BesanÃ§on, France

{bruno.perez,christophe.lang,julien.henriet,laurent.philippe}@univ-fcomte.fr

[2] Service de Chirurgie PÃ©diatrique, CHU BesanÃ§on, F-25000 BesanÃ§on, France

fauber@chu-besancon.fr

[3] Laboratoire de NanomÃ©decine, Imagerie, ThÃ©rapeutique EA 4662,

UniversitÃ© Bourgogne Franche-ComtÃ©, F-25000 BesanÃ§on, France

fauber@chu-besancon.fr

Abstract

Managing the risks arising from the actions and conditions of the various elements that make up an operating room is a major concern during a surgical procedure. One of the main challenges is to define alert thresholds in a non-deterministic context where unpredictable adverse events occur. In response to this problematic, this paper presents an architecture that couples a Multi-Agent System (MAS) with Case-Based Reasoning (CBR). The possibility of emulating a large number of situations thanks to MAS, combined with analytical data management thanks to CBR, is an original and efficient way of determining thresholds that are not defined a priori. We also compared different similarity calculation methods (*Retrieve* phase of CBR). The results presented in this article show that our model can manage alert thresholds in an environment that manages data as disparate as infectious agents, patient's vitals and human fatigue. In addition, they reveal that the thresholds proposed by the system are more efficient than the predefined ones. These results tends to prove that our simulator is an effective alert generator. Nevertheless, the context remains a simulation mode that we would like to enrich with real data from, for example, monitoring sensors (bracelet for human fatigue, monitoring ...).

Keywords: Case-based reasoning, multi-agent system, simulation, risk prediction, adverse events in surgery.

1 Introduction

In hospitals, surgery is performed in increasingly innovative and high-performance settings. Instruments are connected and maintained with all the rigor required by health and safety standards. All parameters are checked before and after the surgery (patient identity and condition, surgical equipment, surgical instruments, etc.) in order to leave as little space as possible for unforeseen events. However, even in this secure environment, minimizing the risk of serious adverse events, incidents, remains one of the main concerns of practitioners [**?**]. As an example, despite all the implemented protocols, J. W. Suliburk recorded 188 adverse events out of the 5365 operations included in their study: 56,4% were due to human error and 51,6% were due to human performance deficiencies [**?**]. Among many other incidents, we can note dosage errors, gestural clumsiness and human fatigue. In this context, we aim at model the operating room environment and resources in order to predict and quantify the patient's exposure to risk. Our proposal is to create a simulator able to reproduce the system evolution, where the system state depends on a set of values, each of which corresponds to the value of attributes such as the vital signs of a patient (temperature, capnia, etc.), the rate of infectious particles or the fatigue of a person. For each attribute, individual alert and incident thresholds are defined by experts. With the simulator, we aim to generate a large number of varied situations and determine their evolution towards incidents. Depending on the risk exposure level, the system state is then classified as normal, alert or incident. ~~The issue we face is to aggregate the values of the attributes in order to identify alert states and thus anticipate incidents. Our study is oriented towards predictive models composed of non-deterministic entities raises, among other things. Our ambition consists of determining alert states which characterize a system evolving toward an incident state.~~ Our objective therefore consists in bringing out risky situations that are a priori unpredictable. In this case, a situation can be considered risky when no alert threshold (individual alert) is reached. A combination of thresholds must then be considered. Consequently, the difficulty resides in the determination of these combined and non-predefined thresholds.

~~despite no individual alert is raised. Basic alert states are triggered when the value of one of our system attributes reaches its associated single threshold. Nevertheless, more complex alert states must also be triggered when values of a set of different attributes are reached. In this partical case, a combination of thresholds must be considered, with combined thresholds values different from single thresholds values. Consequently, the difficulty resides in the determination of these combined and non-predefined thresholds. In this context, our contribution consists in identifying these complex alert states on the basis of acquired knowledge (past experiences). This knowledge is then analysed using decision support tools. Consequently, the multi-agent system is the paradigm we have chosen to generate a large number of situations, some of which are initially unpredictable.~~

~~However, this model shows weaknesses in terms of the acquisition of knowledge based on experience. We therefore propose the coupling between a multi-agents system (MAS) where agents interact in a given environment, and a case-based reasoning (CBR) system that allows solving a target problem by analogy with stored knowledge.~~ In response to these issues, our contribution consists, on the one hand, in bringing out situations with risks that are a priori unforeseeable, and on the other hand in capitalizing on all the simulations (experience base). To achieve these two objectives, we have chosen to couple a multi-agents system with a case-based reasoning, knowing that :

- the SMA paradigm is characterized by agents that interact with each other and produce emerging phenomena;
- the CBR paradigm allows us to solve problems by analogy with past experiences.

In addition to the presentation of this architecture, this paper exposes our work on the classification of different methods of similarity calculations that we consider essential. Indeed, the operating room environment requires a great rigor in terms of the veracity of the information processed.

In this paper, we therefore explore conceptual approaches in relation to the problematic: "risk prediction in a non-deterministic context". The specific literature on predictive modeling (Prognostics and Health Management, Multi-Agents System, Case-Based Reasoning) is the subject of Section 2. We then expose our MAS and CBR architectures in Section 3. Section 4 materializes this architecture with a description of our simulator. Section 5 describes analysis of data management (discrimination between distance calculations). In particular, we present our positioning in response to the determination of aggregated thresholds. We then present our first results in Section 6 and comment them in the next one.

2 Definitions and related work

"Predictive systems" aim to deliver new information from past observations, from simulations or from learning. They mainly depend on the environment and on available data. The era of big data is a major current track in the production of the last ones, which we hope will be both exhaustive and reliable. Several paradigms exist and coexist, either independently or in a complementary manner. Among them, the literature gives great importance to Prognostics and Health Management (PHM). PHM is a major paradigm in the detection of failures for industry, since the impact of an incident is often far-reaching. PHM is a five-step process [?] that starts with data acquisition from sensors (monitoring), proceeds with data manipulation (or processing), then assesses conditions, establishes a diagnosis and finishes with prognostics and decision support.

With this in mind, integrity measures for machining tools, such as cutting machines, prevent the deterioration of work pieces (often expensive) through the anticipation of tools replacement [?]. In [?], the authors show the effectiveness of PHM modeling in identifying risks that are specific to the oil and gas industry in terms of reliability, availability, and maintainability of equipment while achieving cost savings. More and more committees apply the principles of the PHM in the context of human health prognosis [?] with the aim of empowering technologies to provide care. The focus is on biofeedback control of these intelligent technologies for health [?], in a context where a hardware failure can have serious consequences on the diagnosis of human pathologies. However, we did not find any study on risk prediction of complex interactive systems operating in non-determinist contexts such as surgery and the operating room environment. The multi-agent systems described in the following section are built according to the interaction principle and overcome this disadvantage.

Definitions taken from the literature characterize MAS as systems composed of autonomous entities that interact with each other according to certain rules in a specific environment. In [?], Ferber defines agents as autonomous software entities able to perceive (by messages, by data capture...) in an environment. In this context, cognitive agents act with reflection and awareness of their environment composed of other entities, unlike reactive agents that only react to stimuli. Thus, in the case of complex problems such as the epidemiological and ecological analysis of infectious diseases, standard models based on differential equations become unsuitable due to too many parameters and are supplanted by MAS [?]. There is also an increasing integration of the agent paradigm in strengthening collaborative automatic learning that cannot be dissociated from knowledge acquisition. It characterizes the autonomy of the agents and can be implemented beforehand (innate knowledge) or acquired through experience. This second case concerns the entity's ability to explore a knowledge base [?]. Its performance depends mainly on the collection of information that enriches its learning. The CBR architecture designed according to this principle can be a solution to this requirement.

The notion of knowledge acquisition is at the heart of dynamic models enriched by experience. Our modelling of non-deterministic entities is part of this framework and values of the solutions are not necessarily known. It is therefore essential to have a memory of previous cases and an ontology to structure them. Case-based reasoning is an artificial intelligence paradigm based on the development of solutions by analogy with prior experience and general knowledge in the field of application. It is widely used in medicine [?], in industrial maintenance [?] or in stock market analysis [?]. To find a solution, the CBR needs a database of solved cases, the characterization of similarities and finally the knowledge of adaptation processes. Several articles deal with knowledge acquisition, similarity or decision-making with MAS [?]. The different predictive approaches structure their modeling based on the analysis of past experiences. The nature of the data and the analysis tools distinguish these predictive concepts. Two main approaches are considered in the literature for the use of CBR in an MAS. In the first approach, the CBR coupling with MAS integrates, in the agent, the capacity to solve problems from experiments extracted from a case database. In [?] problems are solved locally with sometimes a collaborative approach [?]. In [?], case-based reasoning gives to the multi-agents system the access to a structured database that accelerates data mining. This case search mode, built for a specific problem, however, shows its limitations on non-similar cases. In addition, the CBR/MAS coupling brings a dynamic aspect since it produces a dynamic model enriched by experiments. Indeed, multi-agents simulation remains at the heart of our work, whose challenge is to "predict in a non-deterministic context". The CBR is therefore integrated as a source of active knowledge acquisition and proposes a decision support system in a multi-agent environment. One of the main objectives is to suggest possible answers in different contexts constrained by many parameters. Once again, the value of a case-by-case approach lies in its ability to find solutions by analogy. These can be ordered according to a hierarchy (bargaining agent (BA), expert agent (EA)) [?] or organized into collaborative committees. The system can explore its own case database [?], several merged databases [?], or independent databases. The search for similar cases can integrate different principles of initiation and learning such as artificial neural networks [?]. These decision support tools, which integrate greater adaptability in the acquisition of knowledge, are one of the possibilities we have explored. However, this is limited to case bases, while other types of data such as traces are interesting sources of information.

This type of coupling meets our need in terms of knowledge acquisition, but remains incomplete in response to our predictability problem. Indeed, our model must also integrate the anticipation of non-deterministic situations through an analytical approach of past experiences. As we mentioned in the introduction, our contribution to the integration of this type of analysis uses the MAS/CBR coupling presented in the following section.

3 Architecture of the model

~~This section highlights our originality in relation to the determination of alert thresholds that are not defined a priori. We describe the architecture of our model that integrates the MAS/CBR coupling as a response to our predictive approach. Figure~~ **??** ~~describes the architecture of our decision path.~~

Our research work, which aims to prevent risks in a non-deterministic environment, is materialized (among others) by the implementation of an alert generator. This supposes the possibility of producing a large number of situations as close as possible to those of the studied context (operating room in our case). The MAS paradigm that we have chosen meets this expectation thanks to its ability to highlight unexpected situations. Concerning the acquisition of knowledge, it can either be implemented beforehand (innate knowledge) or acquired through experience. The second case supposes the capacity to explore a knowledge base by the entity. This is why we have integrated a CBR in our architecture whose principle consists in solving a problem by analogy with a problem resulting from past experience. This method also allows to capitalize on experiences. In a schematic way, the coupling between our multi-agent system and a case-based reasoning can thus be defined as a collaboration operated by exchanges. Figure 1 summarizes these. The SMA generates situations (at risk or without risk) which are transmitted to the CBR to be compared with past experiences. The result of this comparison can lead to a modification of the system data (MAS and/or CBR) and generate, for example, an alert (modification of the status of the *alert* agent).

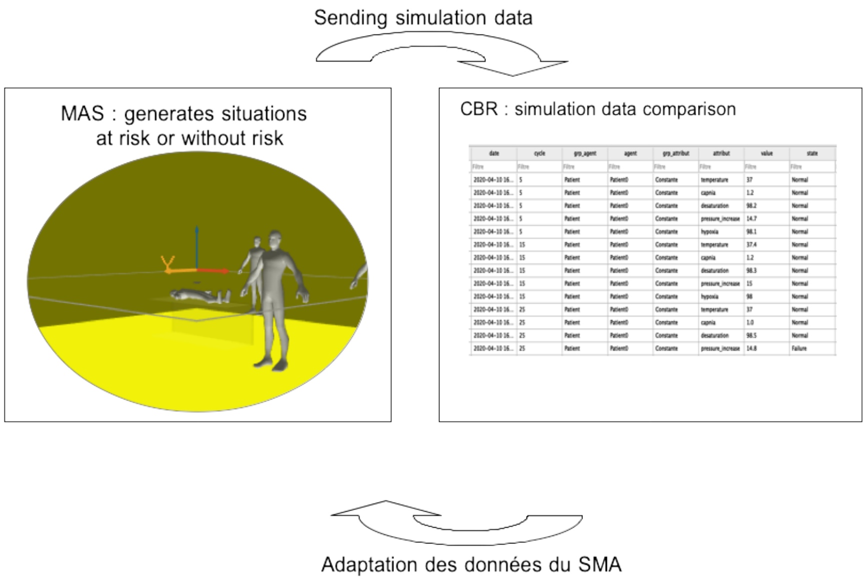


Fig. 1: MAS/CBR exchanges

~~At each cycle (interval between two times t_i and t_{i+1} $i \in [\![1, n]\!]$ equal here to 60 seconds) a new case which takes again the value of each attribute (specific to each agent $a1, a2, ..., a7$) is generated. Each of them enriches a knowledge base (CBR Base) which is then exploited following the CBR steps described in section 3.3. The following section details the architecture of our MAS and CBR.~~ The next three paragraphs describe respectively: the formalism used, our multi-agent system, our case-based reasoning, and finally the MAS/CBR coupling (global vision).

3.1 Formalism

All simulations are sequenced in identical time periods called cycles. Each of them corresponds to a period of time going from t_i to t_{i+1} $i \in [\![1;n]\!]$ which can be parameterized (for example $60s$). The state variables (named attributes) define the behavior of each agent that is essential to achieve the simulation objectives. Their values are modified at each simulation cycle. They give us the state of the system which thus corresponds to the value of the attribute(s) of each agent (the patient's temperature for example) at a given moment (at a given cycle). All these values give a state to the system (global state of the system): $normal$, $alert$ or $failure$. The state $normal$ precedes the state $alert_i$ which is succeeded by the state $failure$. There are three increasing levels of alert (i ranging from 1 to 3) whose intensity is inversely proportional to the time that separates an alert from an incident ($failure$). As far as data acquisition is concerned, we distinguish two modes within our architecture: offline mode and online mode also called 'real-time mode". In the offline mode, the data results from a function or statistical data and is assigned to each cycle. In the connected mode, the data are issued from a dynamic capture: the connected sensors (bracelet for fatigue, electrochemical sensors for infection rate...) and the monitoring devices.. Whatever the mode considered, these sources of information are considered as resources "at the disposal" of the agents. Thus an agent can request one or more instances of a resource (for example the sound level of a sensor) according to his rights. In this article, we only deal with the unconnected mode, which corresponds to the one currently implemented. This formalism being posed, we now approach the description of our two paradigms.

3.2 MAS model for the operating room

In this non-deterministic context, and in order to achieve our objective (predicting operating room incidents), we model the operating room as a multi-agent system. Thus, the operating room is designed as a multi-agent system where each entity is represented by one agent. Each of them, belongs to a group called species (the agent *surgeon* belongs to the species *Personal*). Note that a species can be composed of only one agent (agent *alert* belongs to the species *Supervisor*). This type of model allows to create different scenarios and to simulate the possible evolutions of the state of the system. Moreover, these models, in which agents interact with each other, make it possible to incorporate risks that have been minimized or ignored to date and thus contribute to the optimization of safety in the operating room. For example, simulating the movement of persons (personal agent) allows to determine the level of suspended particles [**?**] which in turn allows to calculate the level of risk. The architecture of our model, which integrates the BDI (Belief, Desire, Intention) paradigm, has five species of entities: *Personal*, *Material*, *Infection*, *Patient*, *Supervisor*. Table 2 gives a description of the attributes of these agents. On an experimental basis, we simulated the evolution of infection sites in parallel with human fatigue and patient's vitals over a period of time divided into cycles.

We chose to model the evolution of the values of the most significant attributes in an operating room according to experts in the field (surgeons and anaesthetists of the Besancon University Hospital). The following paragraphs therefore explain the modelling of human fatigue, that of patient's vitals and that of infection. The major role of the agent *alert* will also be described.

Modeling fatigue. Fatigue can be modelled using several data acquisition methods. In offline mode, data is extracted from statistical files or from a function. If it is a dynamic capture, the connected sensors (bracelet for fatigue) and moni-

Species	Attributes	Comments
Personal	*intention*	operate a patient under optimal safety conditions
	desire	use human and material resources (personal, material)
	belief	measures useful for decision making (monitoring, alert)
	tiredness	fatigue rate (scale from 1 slightly tired to 5 exhausted)
	experience	junior, senior
Material	*function*	hardware functionality
	mat_tiredness	material efficiency (scale from 1 effective to 5 ineffective)
	infected	Boolean
Infection	*type*	type of infectious agent (contaminant, resistant)
	local	has an impact on an area, on the operating room or on both of them
	desire	defined according to its type (contaminant, resistant)
	belief	appreciate the recovery with the future host
Patient	*state*	health status
	surgery_type	urgent, not urgent, complex, not complex
Supervisor	*intention*	prevent a failure
	desire	preventive alert (before the failure occurs)
	belief	listen and monitor the evolution of the data influencing the surgical intervention
	level	alert thresholds

Fig. 2: State variables of simulated agents

toring will provide the information. In our case (offline mode), we have chosen to define the growth of fatigue based on an exponential function (user-configurable) because it can be applied to continuous phenomena and highlights the non-linear nature of fatigue. It is given by the following relationship: $f(t) = ae^{(k\times t)}$ where a is the initial value, k the growth constant, and t the time. It should be noted that in our case we treated the fatigue level of a single agent knowing that each participant may have its own risk threshold.

Modeling patient's vitals. As for the fatigue, Patient characteristics and constants can be extracted from statistical data. However, there are no adapted monotonous functions for this type of variation. For this reason, the data evolve randomly according to a step set by the experts. We choose dynamic data captured from sensors or monitoring in a connected context.

Modeling the infection. The two main types of contamination which can be observed are, on the one hand, exogenous agents in the operating room and, on the other hand, endogenous agents (belonging to the patient). Our system is able to simulate the evolution of the infection rate of exogenous hosts according to decontaminating agents and time. The progression or regression of infection is related to the overlap between contaminated or healthy particles over time.

alert agent The *alert* cognitive agent is central to our architecture and plays the role of:

- central control (gathers the different alert thresholds);
- adapter (manages a collective alert level);
- regulator (proposes possible solutions).

Its preventive role, characterized by the possibility of alerting or adjusting its alert thresholds, requires prior access to the past experiences which characterize the case base in CBR.

~~Other key points of our predictive model include the interactions between different entities (agents). An alert does not only depend on an single threshold but also on the aggregation of different levels depending on each attribute of the system. It implies to think about the definition, the representation and the measure of these collective thresholds. To overcome these two necessities related to alerts and interactions between agents, we suggest coupling our MAS with case-based reasoning.~~ In the introduction of this paper, we have notified the importance of knowledge acquisition in our system. More precisely, it is a question of choosing a paradigm that integrates on the one hand the enrichment of its database from experiences, and on the other hand the search for the solution of a problem by analogy with other problems already solved. Case-based reasoning, adapted to this type of learning, is presented in the following section.

3.3 Knowledge acquisition: case-based reasoning

In the following paragraph, we propose a brief description of the five specific steps of the case-based reasoning cycle.

Elaborate. The monitoring of our system and the resulting decision aids are formalised as follows: $U \rightarrow R$ with U characterizing a sequence of quadruplets (E, A, V, c) and R the couple (attribute value,recommendation). We have assigned the attribute A of value V at cycle c to each entity E. In the example : (nurse,fatigue,1.5,1200) $\rightarrow$ (normal,no_preco), the value 1.5 is assigned to the attribute "fatigue" of the agent "nurse" at cycle 1200. In our CBR system, a case is composed of two parts: the *problem* part corresponds to the quadruplet U and the *solution* part corresponds to the couple R.

Retrieve. The retrieve step is usually based on finding the source case (stored in the case base) the most similar to the target case (case to be retrieved). In our context, we will compare a set of target cases with a set of source cases. Indeed, the system state (MAS) that we record in our CBR database at each cycle, or that we compare, is composed of an identifiable set of cases (for example $\{((surgeon, fatigue, 1, 10), (normal, no\_preco)), ((patient, temperature, 38, 10), (normal, no\_preco))\}$).

Since quadruplets are not necessarily homologous (different attributes), we have chosen to compare each quadruplet of the set of target cases with all sets of quadruplets of the source case. The calculation of the distance of the selected vector is defined according to the relation :

$$sim(\overrightarrow{C}, \overrightarrow{S}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sqrt{\sum_{k=1}^{4} (1 - I)^2} \tag{1}$$

$I = \left(\frac{x_{ki}}{y_{kj}}\right)$ if $x_{ki} \leq y_{kj}$. Otherwise $(y_{kj} \leq x_{ki})$ $I = \left(\frac{y_{kj}}{x_{ki}}\right)$.

x_{ki} and y_{kj} are respectively the values associated with each element of the target quadruplet ($\overrightarrow{C}$) and the source quadruplet ($\overrightarrow{S}$) among n quadruplets. The comparison of quotients exonerates us from the scale problems specific to the different attributes. We consider this distance calculation as canonical and complementary to other approaches defined in Section 5.

Reuse. This step is used to define the target case solution. In the case of automatic processing, the inheritance of the source case solution modifies the target case one. When an adaptation is necessary, (for example with a sufficient but imperfect similarity), we apply the following decomposition:

$$\text{path(srce,target)} = (p_0 \overset{r_1}{\rightarrow} p_1 \overset{r_2}{\rightarrow} p_2 ... p_{n-1} \overset{r_n}{\rightarrow} p_n\ )$$

p_i : system state, r_i : transition between two states (p_i and $pi + 1$)
As an example, let us consider the following case:

$$\text{target } p_0 = (\text{nurse\_fatigue\_limit} \wedge \text{nurse\_dexterity\_threshold}),$$
$$\text{srce } p_n = (\text{surgeon\_fatigue\_limit} \wedge \text{surgeon\_dexterity\_limit}).$$

Figure 3 summarizes this request: the agent *nurse* does not belong to the case base, but it belongs to a parental (*personal*) class with which it is possible to compare the degree of similarity of an agent of the same nature (from the same group). It is therefore possible to associate the collective thresholds inherited from the surgeon class to this new class.

Review and retain. Once experts in the field have corrected it if possible, each new target case, is added to the case base with in order to optimize further searches of similar events.

~~In this section, we have shown that integrating case-based reasoning into an MAS improves automatic knowledge acquisition. Beyond this coupling, data~~

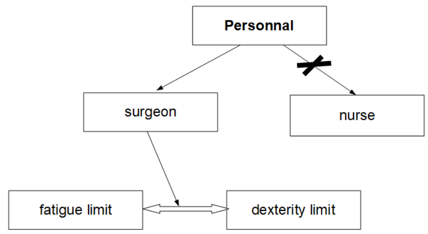


Fig. 3: Adaptation

~~analysis and the quality of data selection are key elements developed in the following section.~~

The global vision of the two paradigms we have just discussed is the subject of the following paragraph.

3.4 Global vision of the MAS/CBR coupling

The diagram 4 provides a global vision of the MAS/CBR coupling. It distinguishes the MAS and CBR paradigms. The multi-agent system is composed of agents belonging to different species such as: $Personal$, $Infectious$ and $Patient$. The state variables described in Section 3.2 characterize the behaviors, actions and objectives of each agent. CBR is represented according to the five cycles (Recall, Reuse, Review, Memorize) described in Section 3.3. MAS and CBR interact in the form of cooperation. This is because the determination of the system status and its updating depends on the reasoning by analogy specific to CBR. In the same way, the enrichment of the case base results from the state of the system exhibited at each cycle by the MAS. Beyond a classical representation, our architecture integrates a holistic organization to our SMA (agent aa interacts with all species), and the possibility for our case based reasoning to interact with several data acquisition sources (data sensors, monotoring ...). These two specificities are not discussed in more detail in this paper.

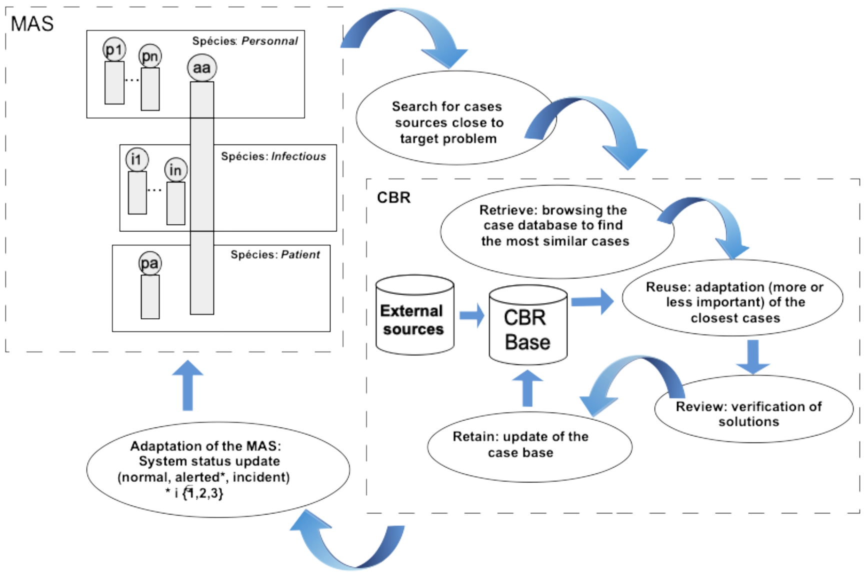


Fig. 4: Global architecture

The following section presents the interface of the simulator, the form used for each type of data by RÃ PC and our first simulations.

4 A similator for risk prediction

We have described in the Section 3 of this paper, our architecture built according to the coupling of a multi-agent system with a case-based reasoning. Our different choices and contributions in terms of structure have been explained. The present section is rather devoted to the exploitation of the model and thus to the generated simulations. We distinguish a visual aspect through the MAS, unlike the CBR which is rather similar to the hidden side of the simulator. The two following paragraphs present respectively our simulator built from the Gama platform and the structuring of the data of CBR.

4.1 Gama platform: a tool to implement the simulator

As shown in Figure 5 the user interface is split into two blocks. The first one is reserved for parameterization and the second one for visualization of the current simulations. In this example, this last one is subdivided into several windows in which evolve (among others): operating room agents, patient constants, infectious agents and system status (visualization of alerts).

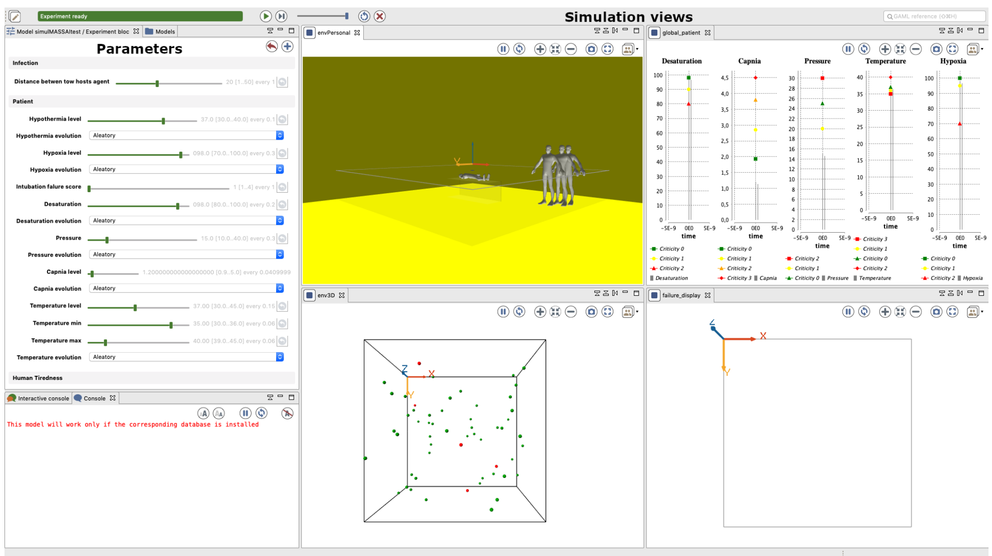
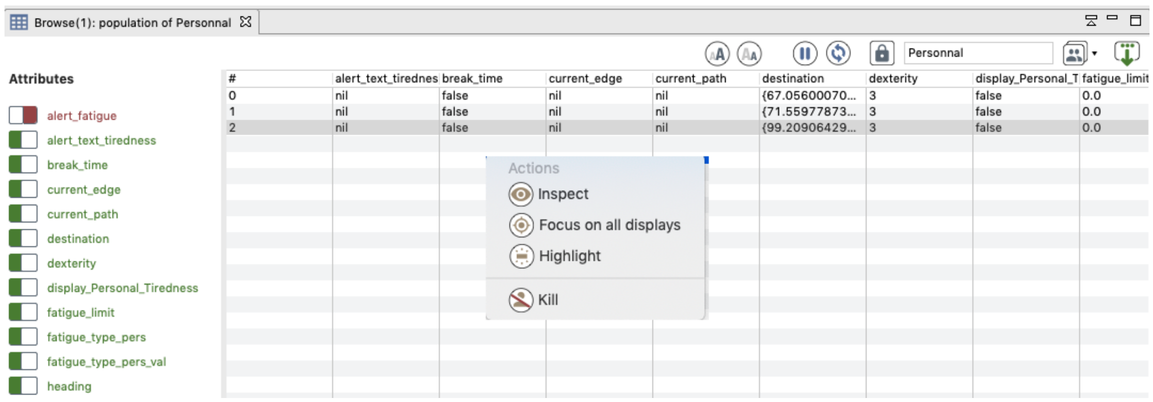


Fig. 5: Global vision of the simulator

The parameterization interface (Parameters) allows to instantiate the alert thresholds specific to the monitored attributes (human fatigue, patient constants, level of infectious risk) and to determine the type of evolution of the values (increasing, decreasing, random). In addition to these parameters, there are also the following possibilities: neutralization of one or more attributes and deletion of one or more agents that you wish to exclude from the simulation. Figure 6 illustrates the possibilities offered by the “Actions” tool of the Gama platform. Each attribute of an agent can be disabled (*alert_fatigue* in our example) and each agent can be excluded from the simulation by a simple deletion ("Kil" option). These features are useful among others to evaluate the interactions between agents.



#	alert_text_tirednes	break_time	current_edge	current_path	destination	dexterity	display_Personal_T	fatigue_limit
0	nil	false	nil	nil	{67.05600070...	3	false	0.0
1	nil	false	nil	nil	{71.55977873...	3	false	0.0
2	nil	false	nil	nil	{99.20906429...	3	false	0.0

Fig. 6: Control of attributes and agents

The visualization interface of the simulation in progress, offers several views of the evolution of the different attributes being monitored. These views take several forms depending on the type of agent being modeled. Thus, practitioners (agents of the species $Personal$) are represented by objects (characters) evolving in the operating room. The evolution of the values of their $fatigue$ attribute is visible individually as shown in the "envPersonal" view. The patient constants are represented by graphs (see "global patient" view) and the agents of the species "Infectious" by a 3D view. The alerts that appear in the "failure display" window indicate their proximity to the occurrence of an incident (far, *near*, *imminent*). This visual approach is legitimate in a simulation context where the aim is to highlight alerts. That said, we have integrated the case-based reasoning and motivated our choices in the previous section. The following section presents the operational mode of CBR and in particular the forms and methods used for the exploitation of the case base.

4.2 Data exploitation: an approach by analogy

The implementation of case-based reasoning and the quality of its exploitation are closely linked to the care taken during the elaboration phase. Indeed, the determination of a case and the indexes used condition the operational efficiency of the CBR. The interactions between the two paradigms impose either the creation of a gateway or the same data structuring. This second option that we have chosen, led us to integrate the state variables of an agent as descriptors of a case while keeping the same data structure. Thus the state variable (attribute) $fatigue$ and the agent *surgeon* are taken up in the same form and correspond to the descriptors A (attribute) and E (entity) of a case. Currently, the case database is enriched from the multi-agent system. The example in Figure 7 represents a set of cases that illustrates the possible states for each of them (*normal*, $failure$). The attributes (*temperature*, *capnia*, *desaturation*, *pressure_increase*, *hypoxia*) of the agent $patient0$ chosen for this simulation have a status equal to *normal* for cycles ranging from 15 to 25. It is equal to $failure$ in cycle 65. It is a detailed view of all the agents and attributes of the system for a given cycle. The state of the system that we recall is driven by the agent *alert*, is a synthesis of all the states of all the attributes observed at a given cycle. Any serious undesirable event resulting from the evolution of one or more attributes, then modifies the state of the system whose value becomes $failure$ for the current cycle. The previous states browsed in an antechronological order take respectively the values: $alert0$, $alert1$ and $alert2$.

Figure 8 shows a $failure$ state at cycle 65, then respectively $alert0$, $alert1$ and $alert2$ for the previous cycles. We observe that the state $failure$ of the system, results in our example from the failure of only one attribute (cf Figure 7 status $failure$ of the attribute *pressure_increase*). The levels of alerts preceding this state, are those which will be proposed by the system during the search for

date	cycle	grp_agent	agent	grp_attribut	attribut	value	state	preconization
Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
2020-03-28 10...	15	Patient	Patient0	Constante	temperature	37.2	normal	no_preco
2020-03-28 10...	15	Patient	Patient0	Constante	capnia	1.1	normal	no_preco
2020-03-28 10...	15	Patient	Patient0	Constante	desaturation	98.2	normal	no_preco
2020-03-28 10...	15	Patient	Patient0	Constante	pressure_increase	15.5	normal	no_preco
2020-03-28 10...	15	Patient	Patient0	Constante	hypoxia	98	normal	no_preco
2020-03-28 10...	25	Patient	Patient0	Constante	temperature	37.2	normal	no_preco
2020-03-28 10...	25	Patient	Patient0	Constante	capnia	0.9	normal	no_preco
2020-03-28 10...	25	Patient	Patient0	Constante	desaturation	98.2	normal	no_preco
2020-03-28 10...	25	Patient	Patient0	Constante	pressure_increase	15.4	normal	no_preco
2020-03-28 10...	25	Patient	Patient0	Constante	hypoxia	98.1	normal	no_preco
2020-03-28 10...	35	Patient	Patient0	Constante	temperature	37.2	normal	no_preco
2020-03-28 10...	35	Patient	Patient0	Constante	capnia	1.1	normal	no_preco
2020-03-28 10...	35	Patient	Patient0	Constante	desaturation	97.8	normal	no_preco
2020-03-28 10...	35	Patient	Patient0	Constante	pressure_increase	15.4	normal	no_preco
2020-03-28 10...	35	Patient	Patient0	Constante	hypoxia	98.1	normal	no_preco
2020-03-28 10...	45	Patient	Patient0	Constante	temperature	37	normal	no_preco
2020-03-28 10...	45	Patient	Patient0	Constante	capnia	1.4	normal	no_preco
2020-03-28 10...	45	Patient	Patient0	Constante	desaturation	97.7	normal	no_preco
2020-03-28 10...	45	Patient	Patient0	Constante	pressure_increase	15.2	normal	no_preco
2020-03-28 10...	45	Patient	Patient0	Constante	hypoxia	98.4	normal	no_preco
2020-03-28 10...	55	Patient	Patient0	Constante	temperature	37.1	normal	no_preco
2020-03-28 10...	55	Patient	Patient0	Constante	capnia	1.7	normal	no_preco
2020-03-28 10...	55	Patient	Patient0	Constante	desaturation	97.3	normal	no_preco
2020-03-28 10...	55	Patient	Patient0	Constante	pressure_increase	15.1	normal	no_preco
2020-03-28 10...	55	Patient	Patient0	Constante	hypoxia	98.5	normal	no_preco
2020-03-28 10...	65	Patient	Patient0	Constante	temperature	37.3	normal	no_preco
2020-03-28 10...	65	Patient	Patient0	Constante	capnia	1.4	normal	no_preco
2020-03-28 10...	65	Patient	Patient0	Constante	desaturation	97.2	normal	no_preco
2020-03-28 10...	65	Patient	Patient0	Constante	pressure_increase	14.7	failure	reduction
2020-03-28 10...	65	Patient	Patient0	Constante	hypoxia	98.6	normal	no_preco

Fig. 7: System Status Detail

analogy between the target case and the source case. The SMA/RÃ PC coupling is an architecture intended to observe the virtual dynamics of a possible representation of reality but also allows the emergence of unpredictable phenomena.

~~In this section, we have shown that integrating case-based reasoning into an MAS improves automatic knowledge acquisition. Beyond this coupling, data analysis and the quality of data selection are key elements developed in the~~

	simul_number	date	cycle	state
	Filtre	Filtre	Filtre	Filtre
32	2	2020-03-28 10...	15	normal
33	2	2020-03-28 10...	25	normal
34	2	2020-03-28 10...	35	alert2
35	2	2020-03-28 10...	45	alert1
36	2	2020-03-28 10...	55	alert0
37	2	2020-03-28 10...	65	failure

Fig. 8: Focus on system status

~~following section.~~ We did not find any architecture coupling MAS and CBR in the context of risk predictability within an operating room. However, the (multi-player) game “3D Virtual Operating Room” [?] is an interesting tool for risk prevention in an operating room that we compared to our simulator. This comparison is presented in the following section.

4.3 Comparison of our simulator with 3D Virtual Room

“3D Virtual Operation Room” is a role-playing game for medical and nursing students. It consists for each player to take the professional role he intends to play. Each of them then finds himself in a virtual operating room with a patient ready to be operated on. This tool allows the player to be confronted with professional situations resulting from real cases of adverse events. A balance sheet established for each player allows to analyze their behaviors. Table 9 summarizes the comparison according to the objectives to be achieved, the targeted personnel, and the operational mode.

On reading Table 9, we can see differences for each of the elements compared. In terms of objectives, the “3D Virtual Room” platform aims to train students while our architecture is designed to simulate the world of the operating room to produce undesirable events that are a priori unpredictable. Next, in the case of the game “3D Virtual Room”, it is the students who are targeted while in the case of our simulator, it is the health professionals already trained or in training.

Comparative elements	MAS/CBR Architecture	3D Virtual Operting Room
Objectives	Generate alerts and identify new serious adverse events	Training students in risk prevention in an operating room
Targeted Personnel	Operating Room Practitioners	Medical and nursing students
Operational mode	A simulator produces a large number of scenarios for the emergence and analysis of serious adverse events that cannot be predicted a priori	Role-playing game where players react to serious adverse events

Fig. 9: MAS/CBR Comparison with "3D Virtual Room"

Finally, the operational mode differs between the two tools, knowing that "3D Virtual Room" involves players in the form of a role-playing game, whereas our simulator makes virtual agents evolve in the operating room environment. In conclusion, our architecture, which makes it possible to alert and prevent new risks, is more of an operational tool, whereas the "3D Virtual Room" platform is more of an educational tool.

The next section is devoted to comparing different similarity calculations. We believe that the quality of the data selection is a key element of our MAS/CBR-coupled predictive system.

5 Evaluation of different similarity calculations

The notion of event has been linked to cases. The CBR cycle followed allows our system to identify alert states based on the combination of attributes values. As explained in Section 3, alerts are weighted according to their proximity to the preceding $failure$, which threshold is determined by the expert (e.g. 40 for temperature). We thus obtain a base containing 3 types of cases:

- $normal$: without alert;
- $failure$: incident;
- $alert$: $alert0$ weight 0 (case $n-1$) $>$ $alert1$ weight 1 (case $n-2$) >,...,> $alertk$ weight k (case $n-k+1$). $failure$ = case n.

An alert is therefore linked to the exceeding of the threshold of a case attribute observed in our case base. Knowing that a case always belongs to a set (cf 3.3), we have a global approach of the alert. It is no longer only attached to a value, but corresponds to an aggregation of values. This overcomes the difficulty of manual and empirical determination of these alerts. In parallel to this case classification, we propose to open the "*retrieve*" step to main similarity calculations we have evaluated:

- Euclidian distances calculations (see Formula 1);
- Confidence interval calculations;
- Complexity Invariance Distance [**?**] (CID) calculations;
- Cosine similarity calculations.

In order to simplify the reading of this article the expression similarity calculation includes the calculations we have just listed. These calculations are described in the following formula with the exemption of the Euclidean distances already described in Section 3.3 (see Formula 1).

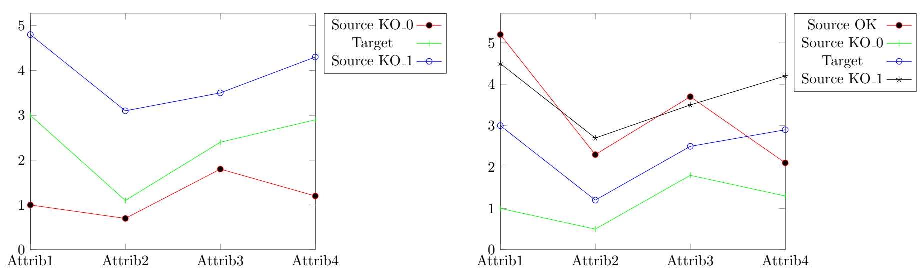
Confidence interval calculations. Using the *state* application previously defined in the "*elaborate*" step of CBR (see 3.3), the formula of the calculation of the confidence value p, for instance for three attributes $X = (x_1, x_2, ..., x_n)$, $Y = (y_1, y_2, ..., y_n)$, $Z = (z_1, z_2, ..., z_n)$, is:

$$p = \begin{cases} \dfrac{\alpha}{\beta} \ , \ \beta \neq 0 \ (cf \ figure \ 10a \ ) \\ 0 \ , \ \beta = 0 \end{cases} \tag{2}$$

with α and β defined by:

$$\begin{aligned} \alpha &= card\{x_i \geq y_i \ \forall \ i \in [\![1, n]\!], \ state(X) = state(Y)\} \\ \beta &= card\{Z \ / \ \exists \ z_k \in [x_i, y_i]\} \end{aligned} \tag{3}$$

Figure 10 illustrates the measure of confidence interval membership. The x-axis shows the attributes and the y-axis shows their values from the source and target cases. Thus, x_i, y_i, z_i, which belong to X, Y, Z respectively, correspond to the y-axis coordinates in Figure 10a. When $p \geqslant ind$ (ind: confidence index initialized or calculated from the evaluation results of the model) any vector belonging to this interval inherits the value of state system ($normal$, $failure$) common to both of the vectors surrounding it. Figure 10a which represents non-secant curves illustrates this type of interval. Otherwise (secant curve as shown in Figure 10b), the system will enrich its learning with distance calculations (euclidian distances d_{ED}, cosinus similarity, CID).



(a) Prediction zone: no secant curves

(b) Prediction zone: secant curves

Fig. 10: Prediction zone

CID distance. The first measure of distance (CID distance) is written as follows:

$$CID(s, c) = d_{ED}(s, c).\frac{max\{CE(s) \ , \ CE(c)\}}{max\{CE(s) \ , \ CE(c)\}}$$

where $CE(x) = \sqrt{\sum_{i=1}^{N-1}(x_i - x_{i-1})^2}$ and N is the number of attributes of each case, s and c designating respectively the source case and the target case. This calculation offers a more robust notion than Euclidean distance, because it includes a corrective factor that refines the similarity measurement.

Cosine distance. The second measure of distances (cosine distance) is obtained by:

$$SimCos(s,c) = \frac{s.c}{\mid c \mid . \mid s \mid}$$

In this calculation, s corresponds to the source case and c to the target case.

The development of a decision tree allows to prioritize these four calculations (Euclidian distance $C0$, confidence interval $C1$, cosine distance $C2$, CID $C3$) in a readable way, unlike neural networks where the predictor is a black box. The classification measurements were constructed according to the C4.5 classification algorithm [**?**] based on Algorithm 1.

Knowing that several modalities are possible (calculates: $C0$, $C1$, $C2$, $C3$) the prioritization rule must be consistent and explicit. In our study, it is related to the strongest representation of modality. The example presented in Table 1 shows the results obtained when applying the 4 calculations which return the state of the source case the most similar to the target case:

$$\forall i \in [\![0,2]\!], \sum_{i=0}^{2} C3_i > \sum_{i=0}^{2} C0_i > \sum_{i=0}^{2} C2i > \sum_{i=0}^{2} C1i$$

$$C1 = \{(normal, normal), (alert_j, normal), (alert_j, alert_j)\} \ with \ j \in \mathbb{N}$$

$$C0 = C2 = C3 = \{normal, alert_j\}$$

Input : R: no targeted attribute set, C: targeted attribute, S: learning data
Output : decision tree
Function *ID3(RCS)*
  if *S isEmpty* **then**
    return simple node value failure ;
  end
  if *S contains identical values for the target* **then**
    return simple node of this value
  end
  if *R isEmpty* **then**
    return a simple node with as value the most frequent of the values of the categorical attribute that are found in records of S
  end
  D ← the attribute that has the largest gain (D, S) among all the attributes of R
  $\{d_j \ with \ j == 1, 2, ..., m\}$ ← the attribute values of D
  $\{s_j \ with \ j == 1, 2, ..., m\}$ ← the subsets of S respectively constituting of records of values d_j for the attribute D
  return *a tree whose root is D and the arcs are labeled by d1, d2, ..., dm and going to subtrees ID3 (R- D, C,s_1), ID3 (R- D, C , s_2), ..., ID3 (R- D, C, s_m)*
end

Algorithme 1 : Decision tree ID3

We applied this hierarchical ranking to five attributes (hypothermia, hypoxia, capnia, infection, human tiredness) and obtained the following result:

$$C3 > C0 > C2 > C1$$

This example shows that we have extended the similarity calculation based

Case	Cycle	state	C0	C1	C2	C3
122	150	$normal$	$normal$	$(normal,normal)$	$normal$	$normal$
123	155	$normal$	$normal$	$(normal,alert2)$	$normal$	$normal$
124	160	$alert2$	$alert2$	$(normal,alert1)$	$normal$	$alert2$
125	165	$alert1$	$alert0$	$(alert0,alert1)$	$alert2$	$alert1$

Table 1: Learning test for the best similarity calculation

only on Euclidean distance. Thus, we have added to the system the ability to discriminate between calculation choices such as CID, cosine distance or confidence interval calculation. This learning optimizes the choice of calculations.

6 Results

We have chosen GAMA [?] as the development framework. GAMA allows to build agent models in an integrated development environment (IDE) including the GAML language (GAMA Modeling Language). The determination of thresholds, which results from a retroactive analysis of an incident ($failure$), can generate an alert when a complex and dangerous situation is reached (proximity detection with a set of thresholds) before reaching each single threshold. Figure 11 illustrates this type of case. Indeed, we can see from the "Infection" and "Personal Tiredness" graphs that their individual $failure$ threshold is not exceeded while a collective alert is triggered (determined based on the similarity calculations that we evaluated and explained in the Section 5).

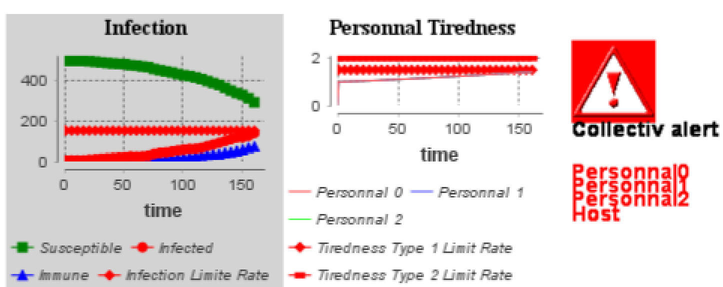


Fig. 11: Fatigue and infection aggregation

In addition to this operational aspect, we compare both of the threshold types (predefined single thresholds, calculated thresholds). In each case, the validity of the status ($alert_i$" $i \in \mathbb{N}$, $normal$) is dynamically affected and verified. In the example of Table 2 we observe that among the 3 cases (case 2049, case 2050, case

2051) prior to case 2052, the CID calculation chosen by our decision support tool (Algorithm 1) is always verified. An approach based on predefined thresholds is only verified once in the same case.

case	status	CID	Predefined thresholds
2049	$normal$	$normal$	$normal$
2050	$alert_1$	$alert_1$	$normal$
2051	alert_0	$alert_0$	$normal$
2052			$failure$

Table 2: Algorithm comparison

Alert detection is determined according to the threshold levels (individual and collective) set during the simulator initialization phase. The curve "Predefined thresholds" in Figure 12 shows that there is no linear relationship between the number of validated cases and time. In addition, the relatively small number of cases validated even after 500 cycles is explained by the difficulty of pre-defining aggregated thresholds in a context where the influences are both multiple and time-dependent.

The determination of the alert depends on distance calculations chosen by screening according to the the evaluation of several similarity calculations described in Section 5. The curve "Calculated" in Figure 12 shows results which are quite close to the reference values.

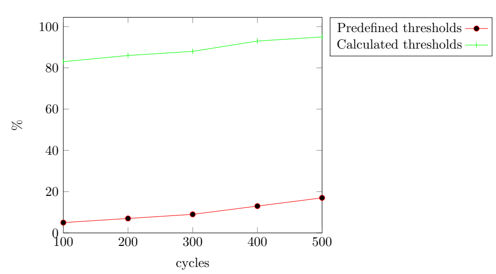


Fig. 12: Prediction zone: secant curves

We applied this same comparison protocol to 1000 simulations and obtained a coefficient of variation equal to 7.1% for the predefined thresholds and equal to 5.3% for the calculated thresholds.

As these results show, the measurement of the coefficient of variation of the standard deviations of each type of threshold (single and predefined versus aggregated and not predefined) for the entire test set (1000 simulations) is low (<15%: the values are homogeneous). This confirms the effectiveness of our results and therefore our position in relation to the MAS/CBR coupling, which allows us to effectively determine the threshold combinations.

7 Discussion

This study shows the effectiveness of the MAS/CBR coupling which allows analytical management of thresholds and avoids contexts where each global threshold is predetermined. For example, the number of possible thresholds for 7 variables would be 127 ($2^7 - \{\emptyset\}$ combinations). In addition, a module for prioritizing similarity calculations (decision tree) has been integrated in order to optimize the selection of source cases. We evaluated four distance calculations (Euclidian distances, confidence interval calculations, cosine distance and CID) knowing that the decision support tool (decision tree) allows us to extend this number to n ($n \in \mathbb{N}$) calculations. This architecture, which allows cases to be generated and then analyzed, is efficient in determining alerts before an incident ($failure$) occurs.

Similarly, this system, which is able to listen to the generated data by a MAS, can be coupled with monitoring systems which evaluate in real time the state of the various entities of the simulator (patient constants, human fatigue, equipment efficiency, etc.). It can therefore be enriched by real situations rather than data from arbitrarily chosen functions (e.g. human fatigue). This new source of values for each attribute will make the case base more reliable. Consequently, the validation by experts of the thresholds determined by the system (aggregated thresholds) increases its efficiency.

One of this MAS/CBR system limitations concerns the anticipation of these alerts and therefore the possibility of making assumptions about the evolution of the system at a given time. Indeed, beyond the automatic determination of alert thresholds, we consider in our perspectives their anticipation by extrapolating the state of the system at time t.

8 Conclusion

This paper presents the MAS/CBR system we have designed and implemented in order to predict the risks in surgery blocks.

The results show that our model can manage data in an environment where the data are as disparate as infectious agents or human fatigue. We have also described our positioning in determining thresholds that are not predictable a priori and therefore difficult to configure since based on combinations of factors. We consider that, in its current form, our simulator is an effective alert generator.

Nevertheless, in a medium term, we plan to integrate our system in a dynamic mode (connection to sensors, monitoring, etc.). We hope that the surgery room environment will benefit from a system where entities are not only reactive but also interactive. Finally, we believe that beyond alert detection (which we have optimized), we could plan a set of possible trajectories from the first cycles based on the state of the system. This is a retroactive study of the drift zones.

Declaration of competing interest

The authors have no conflicts to declare.

Acknowledgements

The authors would like to thank the *European Regional Development Fund (UFC SACD/920/2303/MASSAI-OPE-2019-109)* for their financial support.

References