Accident Simulation: Design and Results

Vincent Idasiak, Pierre David

Accident Simulation: Design and Results

V. Idasiak & P. David
Laboratoire Vision et Robotique, Bourges, France

ABSTRACT: The French legislation regulates the functioning of factories that may be dangerous towards their environment. This legislation imposes the creation of an Internal Operation Plan (P.O.I.) on the plant managers. Those plans describe the proceedings that have to be implemented in case of an accident. Within a framework involving our laboratory and a gas company we have designed a software to create, maintain and execute P.O.I.s . In this paper, in addition to the software functionalities, we present our design method, as well as the used tools. Both are related to the problems of computing the design process of a P.O.I. and its exploitation in crisis situations. The approach is structured around the UML and the © MATLAB tool for both physical simulation and 3D environment management. The final software is developed in Java and uses the API Java-VRML EAI. This solution is currently in use at our partner's plants and it has completely revolutionized their way of managing P.O.I.s.

1 INTRODUCTION

Because of the multiple advantages of simulation and of the breakthroughs made in the improvement of their performances, the use of simulators has spread in a large number of application fields. Indeed, the main qualities of simulations are that they allow low cost repeatable testing of sensitive systems or of systems whose price forbids a full size test. Computer simulation also enables to model systems of high complexity and huge scale, such as the evaluation of natural systems or the study of structures as vast as a whole building. The use of computer simulators mainly prevents people and materials from runnig unnecessary risks, and prevents from engaging expensive machinery, and it enables the repetition of the same phenomenon as many times as necessary. Coupled with the continuous growth in the computing power, computer simulation gives totally satisfiying results in terms of exactness, in comparison to the real observations.

Current simulators have different priorities depending on their missions. This could be the physical accuracy of the results given, or the immersing capacity in the virtual world proposed to the user, by visual and sound aspects or any kind of methods used in virtual reality developments. Currently, the simulators are used for several missions, they evaluate prototypes, train people, predict some natural and physical phenomena or are physical phenomena or are used for purely ludic applications.

Crisis simulation is a domain whose needs perfectly meet the use of computer simulators. Indeed, their goal is often to assess the effects of industrial accidents, natural disasters or spiteful accidents. Therefore, the impossibility of realizing numerous full size tests led to the creation of several crisis simulators. However, the evaluation of the effects is not the only interest of the use of crisis simulators. In France, the legislation concerning plants classified for environment protection I.C.P.E., is based on two fundamental texts (the 19th July 1976 law and it 21st September 1977 decree of application) imposes on factory managers, potentially dangerous towards their environment, to maintain an Internal Operation Plan P.O.I. that describes the procedures to be set up to control the accidents until the plant is in a safe state. This legislation forces the adoption of three principles: creating and maintainming the P.O.I.s, training the personnel to follow the plans and setting up P.O.I.s in crisis situations. That is why we have developed our simulator, called 3D Virtual P.O.I., that meets problems linked to e-learning, as well as the necessity to build a simulator operational in a crisis situation. In this article, we will present the way we developed this software as well as the software as it was delivered to the customer. We listed two great types of architecture used in the design of crisis simulators. Some simulators exploit a geographical database (S.I.G) coupled with a physical engine which runs a mechanical model. The results of the simulation are presented to the user by means of a graphical interface that could be, depending the case, two or three dimensional. For instance, this type of architecture was used for a forest fire simulator. The second class of architecture is composed of an iterative database that communicates with the S.I.G. in order to present the running of the simulation by a multi-view system. This type of simulator is widely used for the simulation of hazardous materials (Hazmat) and for applications such as city crisis. Technically speaking, our tool is closer to the first type of architecture insofar as the S.I.G. is expressed with a 3D world that accurately represents the real studied plants. In our case, the mechanical model is limited to a simple behavioral model. The user controls and communicates with the simulator thanks to a 2D control window and a visualisation window in 3D. With "3D Virtual P.O.I." the setting up of the virtual dynamic world that represents the industrial installations permits to immerse the user in the replica of reality. Thanks to beacons we also obtain the technical information about the functional and security systems referred to in the P.O.I.. It is possible to consult information sheets, to simulate accidents (for example gas leakages, ignited or not, explosions). With the main interface developed in Java, the user can deploy the means of fighting against the disaster. The tool memorizes all the events and decisions and thus builds an accident report or a crisis resolution scenario, in order to improve the initial P.O.I.. The design process of the simulator is based on a method constructed by our team. Within the framework of this contract, we mainly included two stages of this method. In this article we will highlight those two points after having raised up the interaction between the industrial origins of the project and the competences and development of our team. Then, we will detail the characteristics of the simulator. Last, we will present the final tool and the impact of its use on the P.O.I management, before concluding on the work done and the future improvements.

2 DESIGNING PROCESS

2.1 *Industrial issues*

The gas storage sites of the partner company of this project must respect the French legislation about I.C.P.E.s. It was thus necessary for them to equip their factories with a tool, capable of updating their P.O.I.s, but also that could be a precious help in conducting the plans in crisis situations. The product that we had to design was to reach an industrial level of quality for the triple mission of the simulator: teaching, training and the operational management of accidents.

The design of an industrial tool requires (Göransson 2001) to allow the different actors of the project and the future users to intervene at each step of the development process. To minimize its cost, we employed a method centered on the end-user, which led us to apply an incremental method based on the study of a prototype tool (Fig. 1).

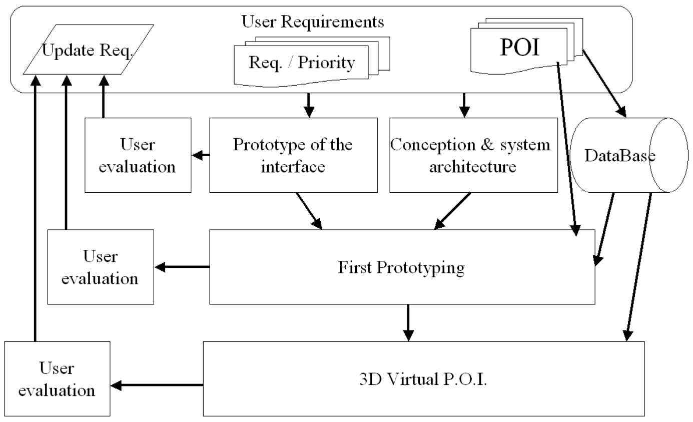


Figure 1. Design process

2.2 *Main developments applied to the project*

Our research team worked on an integrated step aiming at using the support of specification and design teams to provide models to the teams in charge of the reliability studies and reciprocally. It consists in finding shared tools usable by the two specialities to improve the dialogue between them, and thus to create reliable design methods. During the realization of the simulator, the use of this method brought us some help at two levels, that is to say for the implementation of the tool and also for the contruction of the P.O.I.s. The employed method allows to create an application the nearest from the specifications. But it has in addition led to the modification of the specifications and to the evaluation of the modeled phenomenons. This step led us for instance to highlight the need of coherence between each behavioral event and their effects in the 3D interface, and moreover to create the behavioral model of each simulated entity. The deployed method rests mainly on two principles: the extraction of the dysfunctional behavior of the components from a UML functional analysis, and the modeling in © Matlab blocks of the failure modes highlighted in an FMEA.

2.3 *Specificities of the simulator as regards its mission*

We considered the exploitation constraints of this type of software, such as the necessity of having a high power of representation, the necessity of being independent as regards the operating system, to allow the consultation from a CD ROM or a website. Knowing that the sofware should allow: the automatic or semi-automatic addition of information or graphical representation, this without using others phases of development. Therefore we reused works and methods employed for e-learning (Schaerf & Tessicini 2001), (Ewing & Miller 2002) for the design of the static 3D part and the treatment of multi-format information. As far as the dynamic part is concerned, for instance the simulation engine, we adapted former works about remote controlled systems and reliable industrial systems (Benali et al. 2001), (Idasiak & Fontaine 1997).

2.4 *Use and impact of our method on the simulator design*

As the P.O.I. already contains use cases and accident management scenarios, an adaptation towards the use cases and sequence diagrams in UML was naturally done. This allowed us to start the first step of our method, that is to say the synthesis of an FMEA from the functional data found in the sequence diagrams. Traditionally, FMEAs are built from functional analysis. The use of sequence diagrams in order to create an FMEA is totally justified because we observed that it was possible to find the same information about the studied system in both formats. Indeed, FMEAs are used to study the non-realization or the unsatisfaction of the system's functions. The construction of the FMEA is based on the results of the functional analysis because it identifies the functions of the system in the different phases of its lifecycle. Those phases can be found in the sequence diagrams because they describe the different functions in the whole lifecycle. The external elements and their behaviour are also searched; those elements being identified in the sequence diagrams by the entities of "actor" type. The relationships between the actors and the diverse components were also modeled. Finally, the functional analysis makes it possible to identify all the functions of the system, these are the functions which are described by each use case represented by the sequence diagrams. This treatment which creates the FMEA is realised by an algorithm that we developed and whose instructions are described in the following lines.

Input of the algorithm:

- Set of the Actors noted A
- Set of the Objects noted O

$$A \cap O = \varnothing$$

- Set of the messages noted M

The elements of M are couples (x,y) where x, y $\in$ A $\cup$ O, x is the sender of the message and y its receiver.

- C and E are two empty sets
- The set of the failure modes is noted MD

MD = {partial function, no function, intermitent function, unintended function}

Output of the algorithm:

- Table of the FMEA

Operators:

- + is the operator of concatenation on the right
- for m $\in$ M, m[1] represents the first element of m and m[2] the second

Start of the algorithm:

Creation of a table whose first line is composed of 9 boxes that indicate, the name of the object or actor, the failure mode, the cause(s), the effect(s), the severity, the probability, the risk, the means of detection, the technical solutions.

```
$\forall i \in A \cup O$ and $\forall md \in MD$
  $\forall m \in M$, if $m[2] = i$, $C = C + m[1]$
  $\forall m \in M$, if $m[1] = i$, $E = E + m[2]$
Creating a line = {i, md, C, E, $\varnothing$, $\varnothing$, $\varnothing$, $\varnothing$, $\varnothing$
```

The algorithm has been implemented in XML using the transformation language XSLT. XSLT allows to specify how to generate an XML tree (or HTML in our case) from a source tree itself written in XML. In order to realise the specification of the next XSL (norm that includes XSLT) we used rules that associated selectors to a form to be generated (in our case the HTML structure). The selectors are the objects that we search thanks to pointers on the XML nodes. The existing rules on the selectors are setting off the construction of the file.

The use of this method during the prototyping phase (described in section 3) allowed to quickly validate the dysfunctional modes that had an interest for the simulation and highlighted the necessity to add new tools in the method. There is a profusion of dysfunctional modes to study, thus a necessity to create a standard ©Matlab box and a step to introduce the dysfunction in the prototype.

2.5 *Technological solutions*

The interface prototype is based on the analysis of the use cases and the main scenarios. It is divided into two ways of access: the information consultation and the accidents simulation. The functionalities of the information consultation mode as well as the access to technical information or security data and other procedures are totally included in the simulation mode. It quickly appeared interesting to allow an optimised access to this information without having to run the simulation mode, much more expensive in term of resources. The main offered functionalities have been regrouped by the emerging center of interest that we saw in the scenarios and classified by temporal criteria (placing of the accident, settings of the climatic aspects, adding graphic elements …). In parallel, the elaboration of the chosen architecture has been guided by the functionalities accepted by the end-users and the exploitation contrainsts that exists in the crisis management center and in the production plants. A tree based structure of HTML file, generated from an extraction of the database of P.O.I.s, both guarantee the coherence of the information and a fast access to the multiformat documents. The application, accessible from a Web browser, associated with the Blaxxun contact 3D plugin, is deployed with the Java language. The interface with the 3D VRML world is realised thanks to the API Java-VRML EAI (Dit et al. 2003). The advantages of employing a 3D interface in place of 2D views have been discussed in publication like (Cory 2001). The 3D rendering of this product brings the user in an immersing experience of accident scenarios. Thanks to this technology the user understands the geographical aspects of each case much more rapidly than with a simple 2D plan. The 3D environment gives immediately an idea of the installation's dimensions and how the equipments are placed. The details of the viewing window are enough precise to simulate things as the orientation of flames. Moreover the 3D aspects allow to represent easily elements that are far to be insignificant, like the flames high. Finally one plan in 3D shape presents things that would need several 2D plans. For instance in order to represent a gas pipeline, at least two 2D plans are required (one representing the layout of the pipeline and one for the transverse section of the pipe), whereas a single one is necessary in 3D models. The use of 3D view is therefore more effective to enhance the power of representation and the navigation in the virtual environment.

The dynamic simulation part, developed with the formalism of statecharts, is naturally built from the specifications expressed by the collaboration diagrams which merge the different scenarios of each use case. This stage, we also have modeled the dysfunctional behaviors of the diverse objects that appear in the scenarios. We inserted the state that models the dysfonctional behavior thanks to security analysis of the materials as the FMEAs. With the use, they will allow to study the impacts of supplementary accidents or of the unavailability of some equipment during the simulation.

Finally, the software needs to be ready to receive new equipment or to produce new scenarios: a setting at the beginning of the application allows to add some new elements and graphic objects as well as new types of events. Whereas the types of event are simple textual information specific to the job of the end-user, the graphic objects are much more complex. They respect the class diagram of Figure 2, in which we define two types of objects: the simple graphic objects and the event generator objects or objects which have an impact on the physical parameters managed by the simulator. For instance, some of them impact the capacity of fighting a fire by consuming the water reserve. It is a permanent request for the end-user to know the state of the water reserve in real time, and the impact of each means of fire fighting or gas refreshment on this reserve. When we add an object that consumes water the simulator automatically calculates the next state of the reserves. The graphic objects also have proper methods for the visualisation of their dynamic behaviour, their geographic position in the scene depending on the three axis and to the rotation around the vertical axes. The « child shape » class models the tree based architecture VRML and is not managed by the software. By construction it is the VRML engine that realises the graphic operations.

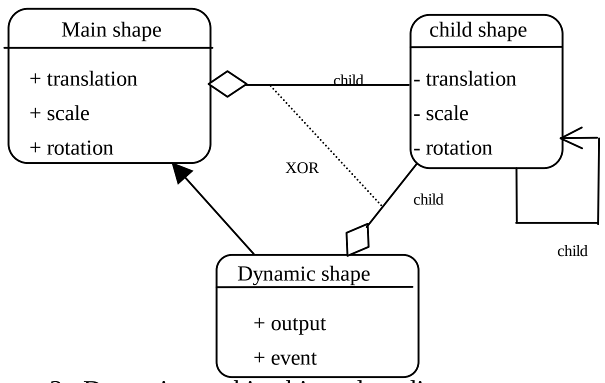


Figure 2. Dynamic graphic object class diagram

It is easy to add some objects. This operation consists in selecting an item in a skeletal list. After a validation, the graphic part is loaded in the VRML file directly in the browser and appears immediately. The insertion position in the interface is the one of the avatar. Therefore, it is a very intuitive command. The static and dynamic attributes are loaded in the Java simulator. The interface described in section 4 offers the user a means to run the application in real time. All modifications of the position and orientation parameters are automatically written in the positioning vector of the VRML file. The modification of the dynamic behaviour involves the posting of the graphic elements (water, color, …) associated with the main object.

3 PROTOTYPING

3.1 *Influence of the prototyping*

For the prototyping of the dynamic part, we have chosen to represent only a fraction of the whole site: the one constituted by the storage area, two storage spheres and three cisterns, plus the area of loading and unloading of the cisterns-coaches. This step uses three formalisms:

The state diagrams model the accident scenario in conformity with the P.O.I., the virtual VRML world is in charge of representing the physical behaviour of the components in order to make the scenarios rapidly understandable by the users. The Simulink part manages the whole simulation, that is to say the interaction with the user, the calculations and the relationships between the two other formalisms and the transmission of the states' values of the stateflow for the animation of the 3D model.

After the validation of the model by the end-user, the Java code of the software has been generated semi-automatically from the stateflow and simulink diagrams. That brought more coherence between the final product and the prototype.

Figure 3 shows the main window of the project in Matlab. The diagram shows the links between the various blocks with the sending of data and events. In a preoccupation with the ergonomics, a single color identifies the blocks related to the same elements (cold colors for water and gas; hot colors for the flames, the temperature and the explosion …).

The stateflow part, Figure 4, saved us a considerable time in the development of the dynamic behaviors to simulate. Note that if we consider the final mission of the simulator only basic behaviors have been modeled.

In order to validate the relevance of our stateflow models, we instrumented them, aiming at memorizing the changings of state of our components on a simulated scale of time. This enabled us to compare the dynamic evolutions obtained with the scenarios expressed in the specifications. We have also generated new scenarios. This property was preserved in the final code, with a list of events.

3.2 *Limits explored and highlighted by the prototype*

In parallel with the project, we have tried out some models of thermal or gas diffusion (Fig. 4) and dysfonctional models (Fig. 4). Some limits appeared, in particular, the complexity to directly calculate, starting from this formalism, a reliability level. We are now translating this work in the AltaRica Data Flow language(Arnold et al. 2002), (Bouissou 1999), in order to offer, for each scenario, a calculation of a success rate. We apply for this work our method that deducts the construction of Mode Automata (Rauzy 2002) from FMEA and functional analysis. With this method we are also able to build Stochastic Petri Nets. All those developments will be dealt with in a future communication. From the FMEA we transpose the failure modes in our simulink model to introduce the main dysfunction (identified by the calculation of the risk) in the system model. We note that this manner of proceeding allows to introduce, since the conception, the impacts of some failure modes and thus to foresee at this step the setting up of mechanism for error recovery or fault avoidance.

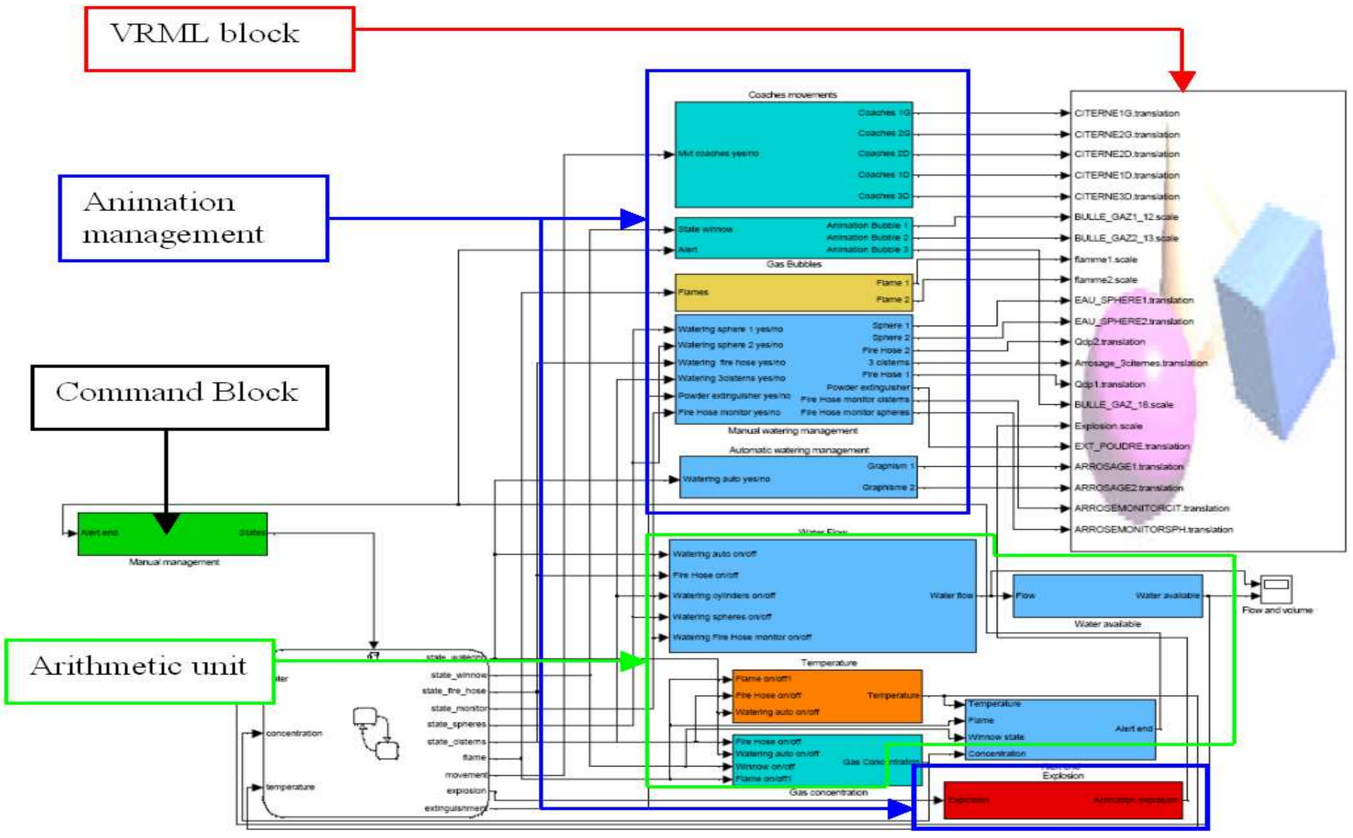


Figure 3. ©Matlab model of the dynamic prototype

3.3 *Prototyping of the graphic aspect*

The animations showing the effects of the gas leakages or the flames are realised by coupling simulink diagrams with the “scale“ parameter of the VRML “shape“ node selected. The simplicity for setting up a cyclic signal generator in simulink makes this operation very fast. We made the effect of bubbles or flames with three groups of elementary shapes, connected with periodic signals out of phase by Π/3. The effect is symbolic, but it is sufficient for the software and really low cost in terms of computing resources.

The interface with the 3D world uses a part of the API Java-VRML EAI. It constitutes a limiting factor for this prototype in the framework of this project. However, if it is not possible or easy to add dynamically 3D objects or to modify parameters such as the transparency, this will be done in the next version of the conception cycle.

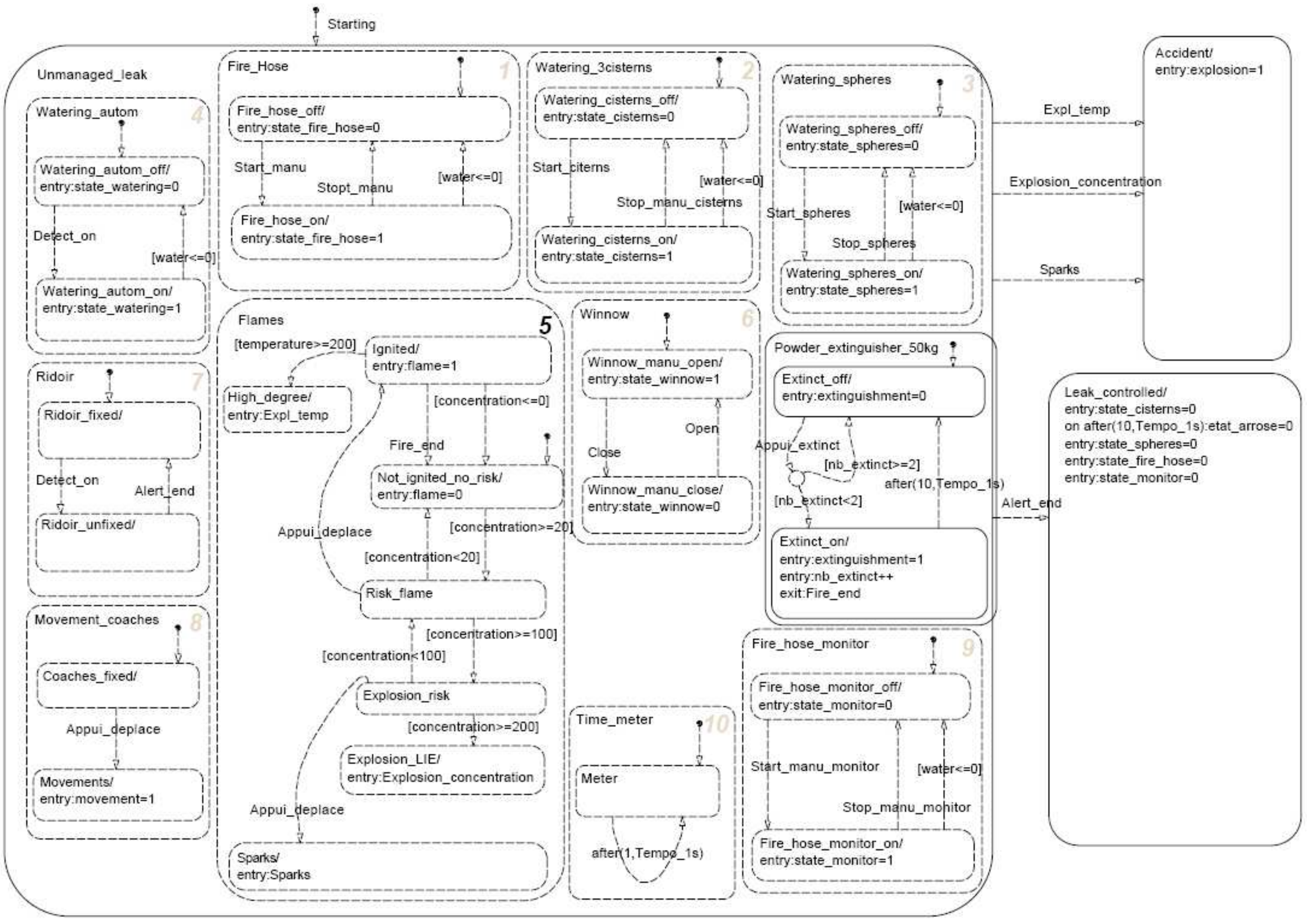


Figure 4. reduced dynamic model of the plant

4 PRESENTATION OF 3D VIRTUAL P.O.I.

4.1 *3D Vitual P.O.I. interface*

This software was developed in collaboration with the TOTALGAZ company, within the framework of a partnership contract, for the topic of computerization of their P.O.I.s.

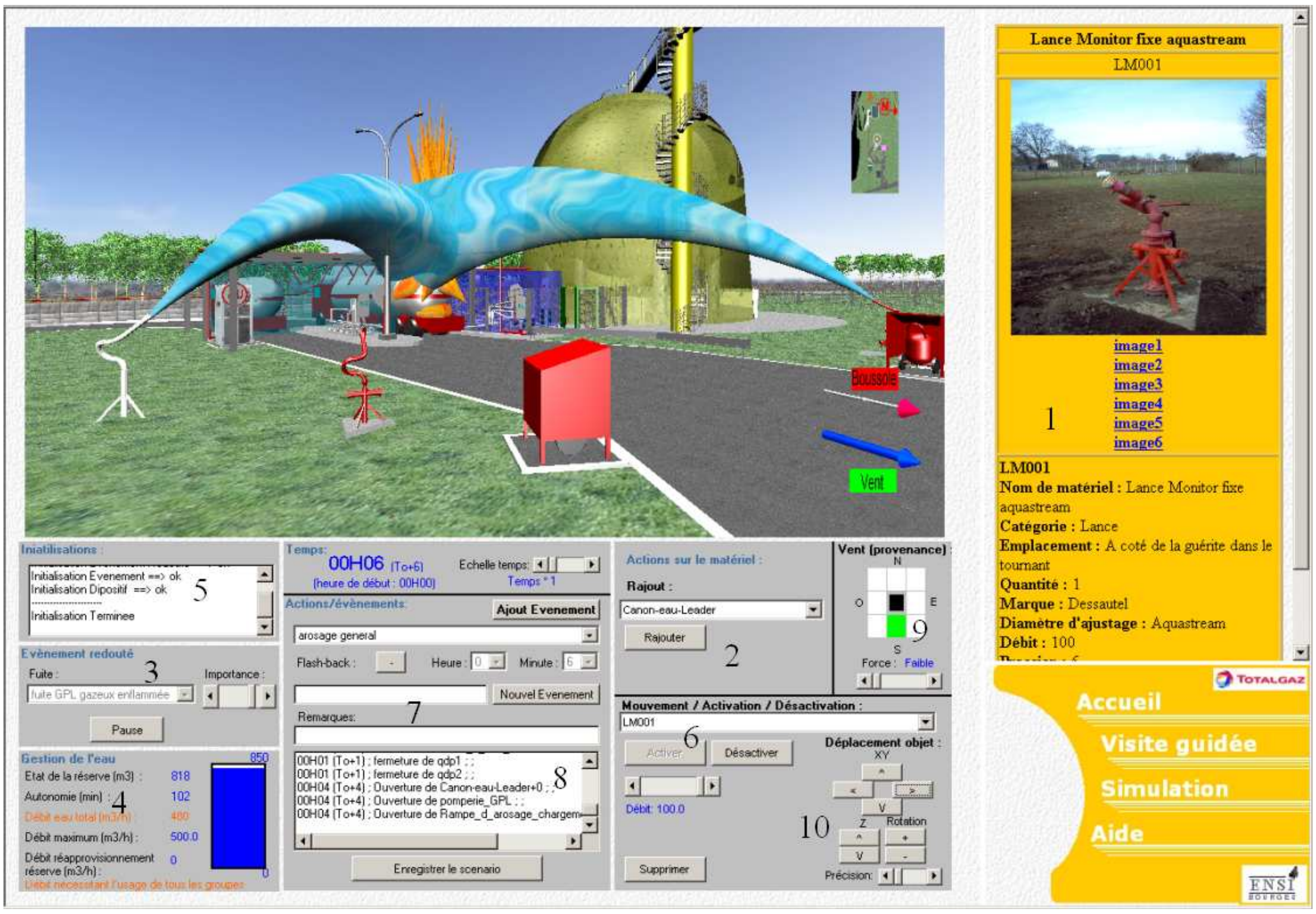
Two types of files are extracted from a data base of P.O.I.s . The first type contains for each object: a single label common to all P.O.I.s, the simulator and the VRML files; the photographs of the real installations and of the mobile equipments; a technical report; finally, technical plans can be referred. Those files are automatically analysed and start the construction of the Web pages. These files appear in the “technical information” window (Fig.5 -①-). Access to this information is very intuitive by the addition of “proximity sensors” on the referred objects. During the movement into the 3D world, if the cursor points a graphic object, a label appears. A simple click starts the posting of information.

The second type of extracted files is essentially composed of a single label and physical caracteristics of the simulation. They are loaded by the Java applet and inform it about the initial parameters of the object that the user can handle.

The interface of the simulator is composed of five panels (Fig. 5);

The “initialization” panel ⑤ gives the state of the loading of the initialization files. They gather the basic information about the simulator, and allow the insertion of the localization of the files of simple objects, event objects and water objects. Thus, all 3D VRML models can be added into the simulator, after a declaration phase. The syntax of the loaded files is verified; no error of parameter file can be accepted. A color indicates a loading error.

The "dreaded event" interface ③ allows the selection of an accident from a list of four possible leakages: ignited gas LPG, not ignited gas LPG, ignited liquid LPG and not ignited liquid LPG. After the choice of the localization and the size of the leakage, the simulation can be run from this panel.



Initialisation Evenement ==> ok
Initialisation Dipositif ==> ok

Initialisation Terminee

00H06 (To+6)
(heure de début : 00H00)

arosage general
Heure : 0
Minute : 6

```
00H01 (To+1) ; fermeture de qdp1 ; ;
00H01 (To+1) ; fermeture de qdp2 ; ;
00H04 (To+4) ; Ouverture de Canon-eau-Leader+0 ; ;
00H04 (To+4) ; Ouverture de pomperie_GPL ; ;
00H04 (To+4) ; Ouverture de Rampe_d_arosage_chargem
```

Fuite : fuite GPL gazeux enflammée

Etat de la réserve (m3) : 818
Autonomie (min) : 102
Débit eau total (m3/h) : 480
Débit maximum (m3/h) : 500.0
Débit réapprovisionnement réserve (m3/h) : 0
Débit nécessitant l'usage de tous les groupes

Rajout : Canon-eau-Leader
Force : Faible
LM001
Débit: 100.0

Lance Monitor fixe aquastream
LM001

LM001
Nom de matériel : Lance Monitor fixe aquastream
Catégorie : Lance
Emplacement : A coté de la guérite dans le tournant
Quantité : 1
Marque : Dessautel
Diamètre d'ajustage : Aquastream
Débit : 100

Figure 5. 3D virtual P.O.I. interface in simulation mode

The "water" gauges ④ indicate in real time all the information about the cumulated flows, the autonomy and the time passed since the occurrence of the accident.

The "event" panel ⑧ memorizes all the actions that occured during the simulation. For each action the time when which they appear is noted. When the applet starts, it is possible to fix the starting time ⑦.

The "device" window ⑥ allows the selection and the dynamic addition ② of devices declared during the initialization. A control block ⑩ provides the means to move the selected object along the x, y or z axis or to rotate it around the z axis. We have also introduced a cursor that permits to modify the importance of the movement in order to improve long displacements and precision in the installation.

The activation, the stop and the dynamic removal of objects are also carried out through the orders available on this panel. Finally, the direction and the strength of the wind are also modifiable ⑨.

4.2 *Impact of the use of 3D Virtual P.O.I.*

4.2.1 *Impact on the P.O.I. document creation*

The P.O.I.s are designed to be utilized in crisis situation that are not yet critical. They are created to limit and describe how to treat restricted accident. Therefore dramatic accident like explosions are not described in Virtual P.O.I because the operation that those cases need are presented in other type of document like P.P.I.s (Particular Intervention Plan), or Red Plans, which were not the goal of the end user. In France the only document which is accepted by the legal institutions to manage P.O.I.s is a document on paper. The use of Virtual P.O.I. leads to the creation of a documentation database more precise and clear. First the software makes it easy to try many scenarios of accident, therefore it is simple to highlight which of them are relevant and to justify the establishment of a formal document made in paper format. By using screen printings from Virtual P.O.I. the formal documents are also more suitable, because they present a realistic perception of the environment impacted by the crisis treated. Finally Virtual P.O.I. has two advantages for the legal document creation, which are the detection and the selection of the scenarios studied on the one hand, and the integration of interesting views and information about the crisis states and resolution on the other hand.

4.2.2 *User experience and reuse*

The software uses very instinctive functionalities for the establishment of the scenarios. Moreover the navigation in the interface and the placement of new objects is very intuitive. All those aspects make the software very easy to be used. After roughly twenty minutes of explanation a new user can start to simulate a first accident scenario. The software has two main functionalities in training, the first is to allow to discover the plant, and the second to conduct accident scenarios in it. Thanks to the fact that the navigation system is very standard, a simple visit of the plant which is very useful for future workers, can be made without experience of the software. The software is therefore easy to deploy for a mission of training. To use the software to follow a real crisis development and to use it as a tool for decision-making aid, a preliminary experience of the software would be very useful.

The direct reuse that we could imagine for our simulator are in the field of the management of P.O.I.s for oil, chemical or petrochemical industries. The design methods and the techniques that we used can be adapted to represent special phenomenon for those technologies. For instance if we would like to reuse those works for a transfer in oil industry, we should proceed identically. A preliminary risk analysis should be performed to identify the accident scenarios and their impacts. Therefore new dangerous phenomenon should be highlighted like boilovers, bleves, toxic smokes or large fires. Heating effects should be supported by colorizing the environment of flames respecting a value scale, solid remains should be added after an explosion by a simple object creation. Those remains could have the effect of a physical barrier and could also carry out a heating information. Liquid leakages (Ignited or not) should be also represented and indication about the maximum flow and quantity of lost oil could be integrated.

5 CONCLUSION

This article presented the options chosen for the development of the simulator. This has highlighted the particular technological choices that we have employed to solve the problem of P.O.I. management, but it has also raised up the use of a general design step stressing the taking into account of the dysfonctional behavior. This method, that aims at integrating the dependability analysis in the conception operations, will be the topic of a future communication in which we will better underline the methodological aspects. The simulator that we presented has shown capabilities for its mission of managing P.O.I.s in gas plant, but it could be deployed for more types of industrial plants or public buildings. It is in reality a tool for testing and setting up emergency plans for any kind of situation. Our current work about this tool consists in trying to introduce smart agents in the simulations, in order to model technicians or firemen that follow orders to solve the crisis, or just panicked people.

REFERENCES

Arnold, A. & Griffault, A. & Rauzy, A. & Point, G. 2002. The AltaRica language and its semantics. In Elsevier, *Reliability Engineering and System Safety*, vol78, 1-12.

Benali, A. & Idasiak, V. & Fontaine, J.G. 2001. A design of Adaptative Teleoperation Application. *10th IEEE international workshop on robot and human interaction, RO-MAN’01, Bordeaux-Paris*: 306-312.

Bouissou, M. 1999. A benchmark for availability. *Reliability & performance evaluation.* France:EDF

Cory, C.A. 2001. Utilization of 2D, 3D, or 4D CAD in construction communication documentation. *Fifth international conference on information visualization, IV’01, London:*219-224.

Dit, S.L. & Degrande, S. & Gransart, S. & Chaillou, S. & Saugis, G. 2003. 3D technology for the Worl Wide Web. Eighth c*onference on 3D Web Technology, Saint Malo, France:* 135-146.

Ewing, J. & Miller, D. 2002. A framework for intelligent Virtual Training Environment: the steps from specification to design. *Educational Technology & Societ*, 5(4).

Göransson, B. 2001. Usability Design: a framework for designing usable interactive systems in practise. Licentiate thesis, ISSN 1404-3203, *Department of Human Computer Interaction, Information Technology,* Uppsala, Sweden.

Idasiak, V. & Fontaine, J.G. 1997. Safety and reliability in the design of computers network for manufacturing. *Reliability’97, institute ofmachine reliability of National academy of sciences,* Minsk, Belarus.

Rauzy, A. 2002. Mode Automata and their compilation into Fault tree. *Reliability Engineering and System Safety*, 78: 1-12

Schaerf, M. & Tessicini, A. 2001. JubilEasy: Build a personalized 3D visit of Rome. *Web3D conference*, Paderborn, Germany.