Collaborative process design for Mediation Information System Engineering

Sébastien Truptil
Université de Toulouse – Ecole des Mines d'Albi-Carmaux
truptil@enstimac.fr

Frédérick Benaben
Université de Toulouse – Ecole des Mines d'Albi-Carmaux
benaben@enstimac.fr

Hervé Pingaud
Université de Toulouse – Ecole des Mines d'Albi-Carmaux
pingaud@enstimac.fr

ABSTRACT

To reduce a crisis, heterogeneous actors must coordinate their actions and exchange information. The ISyCri project aims at facilitating this collaboration by providing a Mediation Information System (MIS), which change the set of partners into a system of systems. The design of this MIS is based on the characterization of the crisis and services of actors. The first step of MIS design consists in deducing a collaborative process involving partners of the crisis reduction (from the characterization of the crisis and services of actors). This step is based on a metamodel, which allows to build models (consistent with each other) and ontologies. The inference of the collaborative process is not a trivial issue: the deducing approach uses ontologies and models transformation to organize services according to characteristics of the crisis. This paper discusses this global approach and an illustrative case of study.

Keywords

ontology, metamodel, collaborative process, crisis, system of systems

INTRODUCTION

As described in (Benaben et al.,2007), the ISyCri project has to deliver a solution to efficiently support organizations involved in mitigation of a crisis. The relevant system must be able to make many information systems interoperate in order to satisfy the needs of the partners involved in the crisis management. More specifically, one purpose of the project is to define an engineering approach which provides an information system adapted to crisis management.

Generally, each actor owns its own information system adapted to its own needs, resources and processes. That is why, we believe, as shown in (Bénaben et al, 2007), that the major issue of integration of partners in crisis context is the ability of these information systems to communicate, to exchange information, to share services and to coordinate their behaviors according to the global goal of crisis reduction. This observation is in accordance with the concept of interoperability. According to *InterOp*[1] , Interoperability is "the ability of a system or a product to work with other systems or products without special effort from the customer or user" (Konstantas et al, 2005). In a first part of the paper, we try to show that the concept of system of systems (SoS) is an appropriate framework for the system adapted to crisis management.

In the second part, a Mediation Information System (MIS) is defined as a candidate solution. It can be seen as a global interoperability service utility given by a software component inside the SoS architecture. In the third part of the paper, we focus on the first step of the MIS design before illustrating.

---

[1] InterOp is a European Network of Excellence (NoE) dedicated to Interoperability issues

CRISIS MANAGEMENT REFERS TO A SYSTEM OF SYSTEMS

Nowadays, many organizations have to work more and more rapidly and efficiently with other actors in their environment. Very often, the collaboration is only effective during short periods of time (whilst, before, they were used to collaborate only after improving their reliability) with correlation to either high returns on investment or low transaction costs. The emergence of this context was illustrated in (Bénaben et al, 2007).

As an effect on engineering activities of this, systems now should be designed to work rapidly and efficiently with other autonomous systems. The concept of system of systems begins to appear as a generic response given by systems engineering to these situations.

Although the concept emerged in the nineties, there is not yet a total agreement about its definition in the associated community of experts. It is possible to emphasize some of the criteria which allow to make the difference between a system and a SoS. Like explained in (Sage and Cuppan, 2001, Maier,1998), SoS exists when there is a predominance of the following properties: operational and managerial independence, geographic distribution, emergent behaviors, and evolutionary development.

In response to a crisis, many actors are concerned which are not necessarily supposed to be hierarchically linked to the main executive unit. The heterogeneity of partners is generally recognized as a real fact in such situations, being a concrete limitation for the management. Thus, the crisis management system is regarded as a set of several autonomous systems that have to collaborate, either for decision making or for operations.

Depending on the type of crisis and its level of expansion, many systems can interact during the crisis management. By the way, we should consider that they could work at different locations. Information and Communication Technologies (ICT) are in charge of information exchanges between the partners, have to put them at work by providing SoS infrastructures.

The dynamic nature of a crisis has many implications for management. The pool of partners can have a different composition during time. Priority on objectives can change. Non predicted processes can be launched depending on the degree of gravity of the crisis, and the decision making of its management. So the evolutionary development of SoS is quite evident.

Finally, if partners interact to be globally coordinated, the expected result is supposed to be better than if they were not. Consequently, the crisis management has probably an emergent behavior.

Thus, most of the characteristics that discriminate a SoS from a system also match with our subject of concern. By consequence, we found appropriate the idea to address the information design problem as a system of information systems design. We will consider partners having autonomous information systems that want to efficiently interact. We try to develop an architecture of this information system which gives the opportunity to self adapt to changes of the context, and respect the many phases of the system life cycle. The final result shall provide an added value related to the crisis management objectives.

Following a system engineering philosophy, we have first studied a functional architecture of our information system. Then, we have defined the constraints related to partner's IS profile. Finally, we have thought about the adaptation of the system to changes using a model driven approach of the design that in some way facilitates closed connections between many architectural layers.

A MEDIATION ORIENTED SOLUTION

A MIS is a candidate to solve the problem under study. It is a system to which data exchange, as service invocations and process orchestrations could be delegated. It is a key design element under the hypothesis that partners are able to expose their services and could be easily connected in an SOA environment. It is adapted to cases for which system interoperability cannot be achieved by a simple assembly of many services because system requirements are not really stable and level of heterogeneity is too high. A MIS is an intermediate component that. could be perceived like a media ability that supports collaboration between partners. By the way, it is supposed to play many distinct roles: translator for data interchange, broker for middleware software connections, coordinator for cross organizational process executions. The emphasis is put on this particular function in the following.

As explained in (Truptil et al, 2007), a MIS design approach could be conceived using a model-driven engineering process, i.e. a dive across many abstraction level (business, logic and technologic), to make interoperability between several information systems of actors becoming a reality, even if the demand is rapidly changing. For doing so, the structure of a MIS component is previously known (Touzi et al,2008), it could be specified by an UML profile, and its implementation could be tracked by many different models from the business point of view to the technical one.

GENERAL APPROACH OF MIS DESIGN

In the case of a major crisis event, the workload necessary to adapt the SoS to a reconfiguration demand must be minimized. One way to satisfy this criteria is to use a framework where MIS engineering tasks are defined and supported by computer facilities. A model driven engineering approach is recommended to enhance the automation of tasks. A second way is to check the level of knowledge on the SoS. Knowledge reuse appears as a capability whose promotion will probably give some benefits. So, considering one particular crisis, the idea is to examine as soon as possible the services that partners offer to consolidate a crisis mitigation scenario. The first step of the approach uses both available information about the state of the crisis, on the one hand, and services exposed by actors, on the other hand, to instantiate a specific domain ontology. This ontology enables a characterization of the crisis, and reasoning on this characterization by using rules. Such mechanisms help the management to set up a cross organizational, or collaborative, process for crisis mitigation (referring to the set of available partner's services). A knowledge based approach is therefore localized at the upper part of the CIM level in a MDA model.

Once this collaborative process is accepted by the actors as a coordination act, a second step is performed which is a transition from business analysis to IT analysis. A model transformation is applied to the collaborative process model in order to deliver (at the PIM level of MDA) a UML based specification that explicitly links the mediation service utility to the partner's services. Finally, using this last model of the MIS, a second transformation is executed and delivers (at the PSM level of MDA) a configuration profile for the target platform. This is the final point of one complete iteration of the engineering process for one configuration of the system.

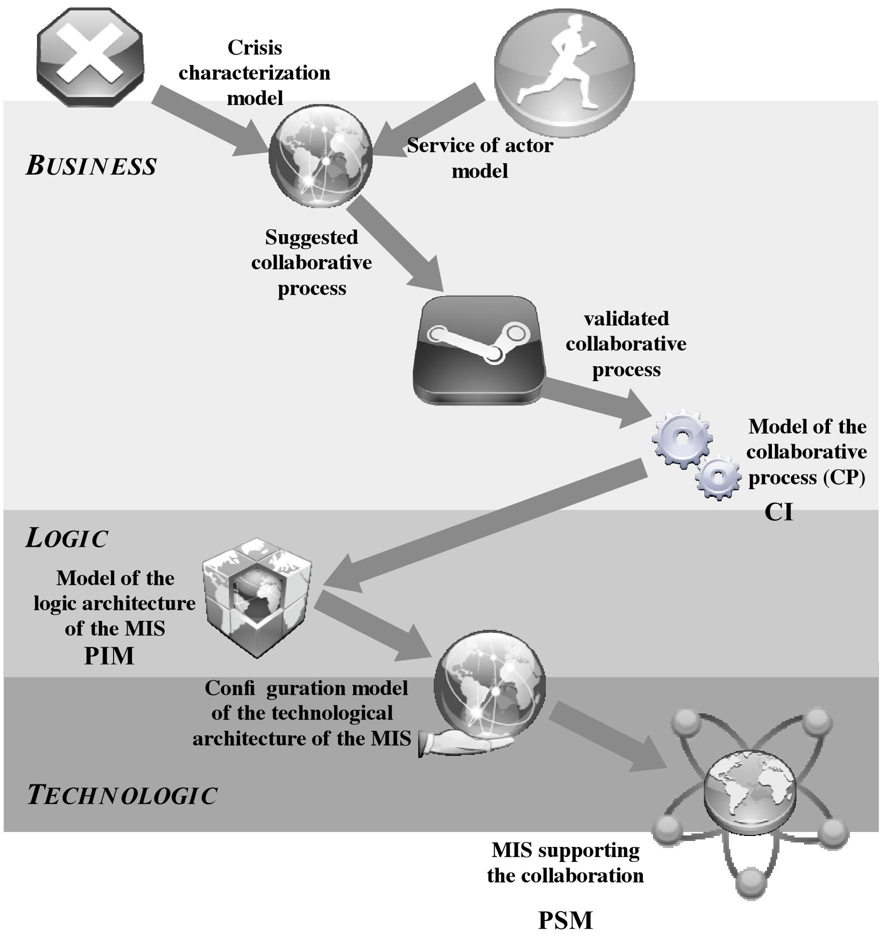

Figure 1. Overview of the MIS Engineering

Figure 1 resumes the main steps of this particular design process with a three layer architectural representation: business level, logical level, technological level. Concerned models and performed operations, while crossing these layers, will be described in more details in the next paragraphs.

Then, a focus was made on the first step of the MIS design approach, which consists in the deduction of a collaborative process from the crisis characterization and partner's services definition.

OVERVIEW OF THE COLLABORATIVE PROCESS DEDUCTION

As mentioned before, the first step of the MIS design approach begins with the collaborative process deduction from the comparison between crisis characterization model and model with the registered services of actors. The first step consists in defining these models thanks to tools: *IsyCrisisTool* and *IsyServiceTool*. Once these models created, they are injected in an ontology: *OntoServiceState*. After that, an inference on a set of deduction rules offers a selection of the services which can be used for the crisis response. Then all the available services are proposed to the user thanks to *IsyServiceState*. This tool is used to define the priority order of treatment of risk and consequence and services which will be used for the crisis response. This group of services is injected into the *OntoProcessDeduction* ontology. The aim of this ontology is to define the execution order of services. Finally, the collaborative process is modified by user thanks to *IsyProcessTool*. At the end of this approach, the collaborative process is ready for the step of the validation.

SIX-STEP REASONING METHODOLOGY

This part explains the several steps of the collaborative process deduction as well as the tools which are used.

Step 0: Before the crisis emerges, it is necessary to create a model of definition of partner's services. This model must define all the conditions of use of services, *i.e.* which risks are prevented, which consequences are reduced by this service and what can forbid the use of such service.

Step 1: When a crisis is declared, a model composed of all the data about the crisis is created. These data are composed of components which are impacted by the crisis, the trigger event, the consequence and the risk of the crisis.

These two steps consist in creating models. Therefore, two modeling tools are used: *IsyServiceTool* and *IsyCrisisTool*. GMF (Graphical Modeling Framework) (GMF) is used to create these tools.

Step 2: Once the models are created, they were injected in an ontology created with Protégé (Protégé, 2000): *OntoServiceState*. The injection is realized thanks to style sheets.

Step 3: Once the models are injected into *OntoServiceState*, deduction rules (later explained on this paper) are applied for the determination of services which can be used for the crisis response.

Step 4: Available services are extracted from the ontology and injected into *IsyServiceState*. This tool is used to allow the user to define the priority of treatment of the risk and consequence of the crisis, and to select the services which are used for the crisis response. This tool is composed of two parts: (i) a general menu, which resumes all the risks and consequences of the crisis as well as their state. The State can be *KO* if no service, which can solve this problem, has been selected. The state can be *Okcan* if one or more services which solve this problem are selected. The state can be *Okcould* if one or more services can solve a problem close to the crisis' problem has been selected. (ii) for each problem, all the services which can solve the problem are visualized in two categories: *can be done,* services which solve the problem exactly, *could be done*, services which solve a problem close to it.

Step 5: Once services selected are and the order of priority of treatment determined, these information are injected into the *OntoProcessDeduction* ontology. The aims of this ontology is to define the order of execution of the services. This ontology will be described in this paper.

Step 6: The collaborative process model is proposed thanks to *IsyProcessTool*. This tool allows the user to modify the process.

Finally at the end of these steps, the collaborative process is deduced. However, we can notice that this approach is based on a comparison of models. It is thus necessary that all the information modeled in various models remain coherent. A crisis metamodel was created for this purpose.

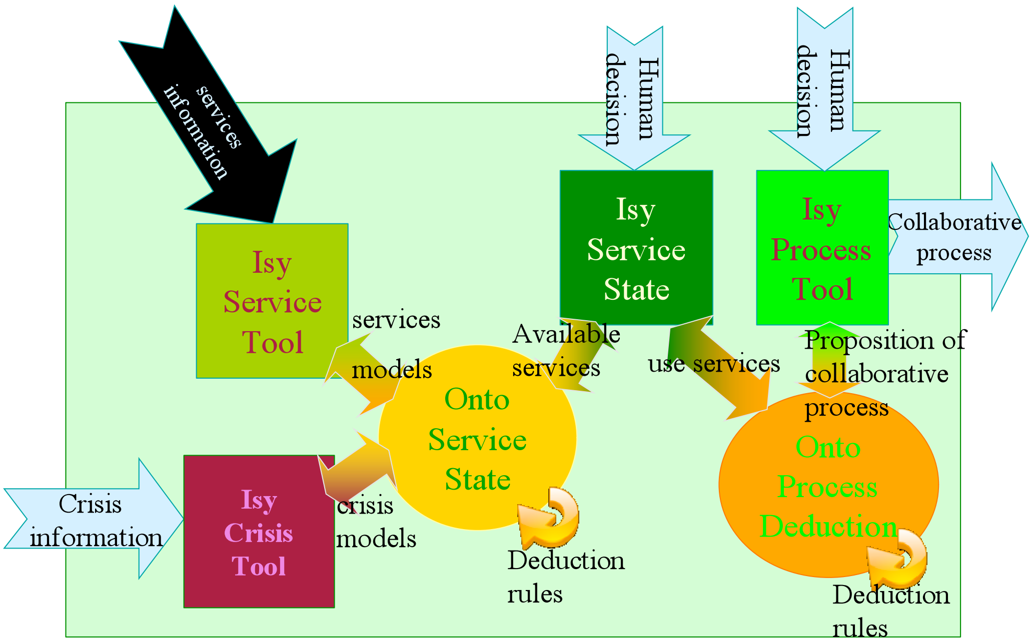

Figure 2. Overview of the collaborative process deduction

Figure 2 resumes the steps of the collaborative process deduction. In this figure, ontologies are represented by circle, and human interfaces are represented by square, the arrows show the information which are transferred between tools or which are transferred from human to tool. On this figure, we can find the six step if we read it from the left to the right.

CRISIS METAMODEL

This part describes the crisis metamodel. This metamodel is used to certify the coherence of models created as explained in (Bézivin, 2004) and (Luzeaux et Ruault, 2008). This metamodel, described in UML, has been built on the basis of the capitalized knowledge collected from several crisis (especially civil and humanitarian crisis) and about risk management. This metamodel is composed of four main parts which are: *studied system*, *crisis characterization*, *treatment system* and *collaborative process*.

·The studied system:

The studied system is defined as the sub-part of the world affected by the crisis. The components of this subsystem have been grouped in different categories such as goods, natural sites, people and civil society. All these elements are considered as studied system components which can be concerned by the situation.

Goods are like each man-made entity (roads, bridges, buildings, houses, etc.). On the contrary, natural sites are the elements of the studied system which are not man-made, such as rivers, forests, etc. People concern all the groups of persons which are threatened by the crisis situation (people from a city, group of travelers, employees of a company, etc.). Civil society includes legal entities (media, intellectuals, etc.), associations and organizations which act in the crisis area.

The studied system also contains risks and characteristics. A characteristic is a particularity of the sub-part of the world affected by the crisis and one or several risk(s) may concretize the exposure to this characteristic. For instance, an area like Japan presents a characteristic of seismic instability while an earthquake occurrence is a risk attached to this characteristic.

An event can realise risks and thus create consequences. Moreover one or several events can trigger a crisis.

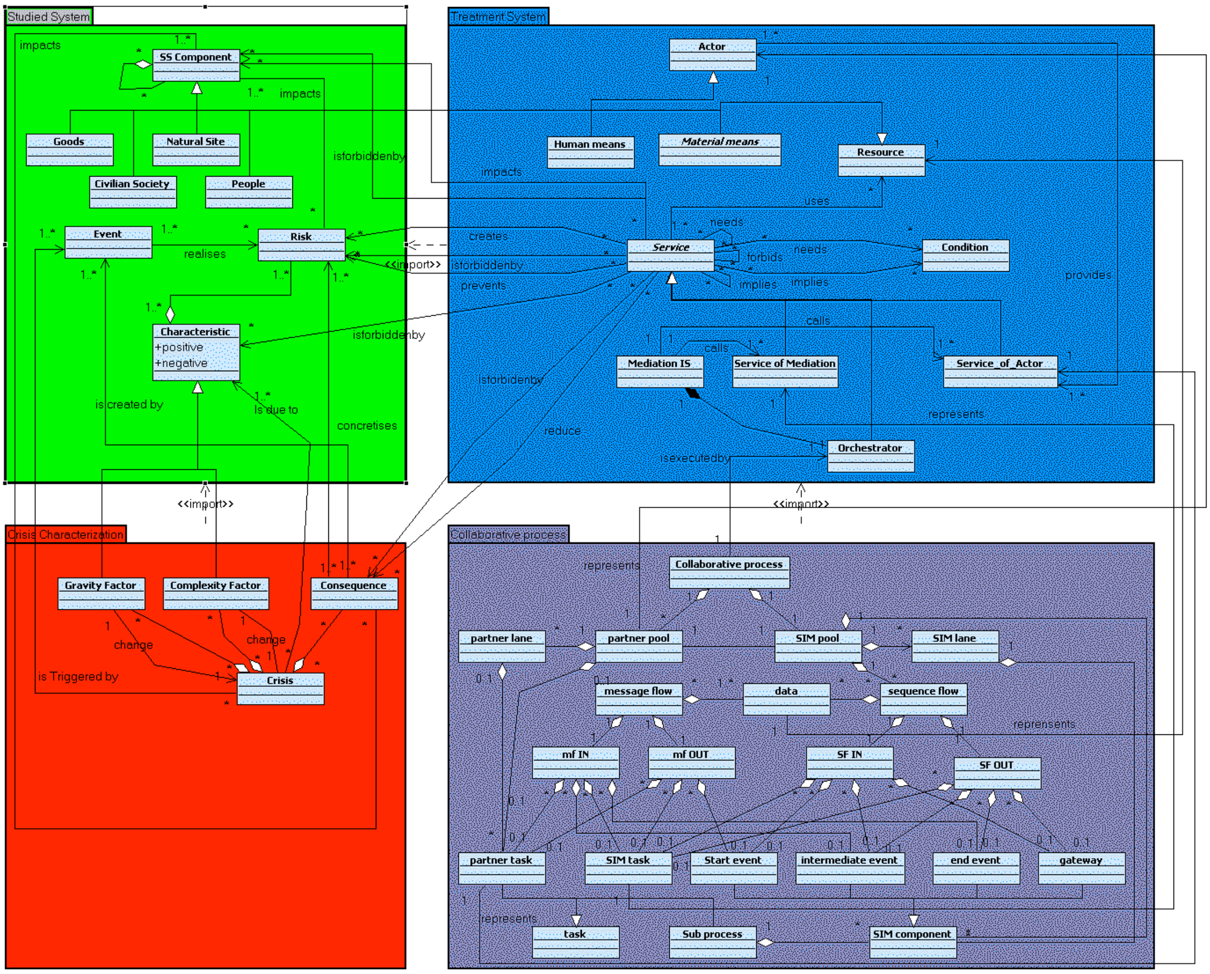

Figure 3. Crisis metamodel

·The crisis characterization:

Once appeared, a crisis is composed of three main components: (i) consequence(s), (ii) complexity factor(s) and (iii) gravity factor(s).

A consequence is the noticeable concretization of one or several risks on the crisis studied. A complexity factor is a characteristic which impacts directly the nature of the crisis and can affect its type (for instance, a sanitary crisis may evolve into a social crisis due to the "over-communication" through the media). A gravity factor is a characteristic which impacts directly the gravity of the crisis (for instance, a strong wind and a dry weather could affect the gravity of a fire in a forest).

·The treatment system:

In order to solve (or at least to reduce) the crisis situation, it is necessary to define a treatment system, composed of services, which aim to drive the situation to a stable and under control state.

A service, using resources (human or material), is used to reduce a consequence, to prevent the concretization of risks or to lead up the system to a condition, special state of the system, needed by another service. However a service can be forbidden by risks, consequences, another services or a studied system component. For example, a risk of building collapse forbids the service of sending firemen inside.

There are two kinds of services: (i) service of actor and (ii) service of mediation. A service of actor is a service provided by an actor, the authors make the assumption that all their information systems have the same conceptual logical model: Service Oriented Architecture and all the services provided by an actor are services of its information system. Unlike a service of actor, a service of mediation is not provided by an actor; indeed, service of mediation can be a service of coordination between two services of actor or can be a value added service, like a web service of weather information or like an orchestrator, which enables the execution of the collaborative process.

·The collaborative process:

This part is dedicated to the description of a collaborative process (it includes elements of process modeling) and is directly inspired from a metamodel of collaborative process described in (Touzi et al, 2008).

ONTOLOGIES

In this part, we present ontologies: *OntoServiceState* and *OntoProcessDeduction*. Then for each of these ontologies, the reasoning of deduction will be explained step by step.

An ontology is a specification of a conceptualization (Gruber, 1993). It contains a set of relevant concepts of a field, their definitions and their links. This is why these two ontologies are based on the metamodel presented previously. *OntoServiceState* aims at deducing the available services to the crisis response. For that, it is based on the *studied system*, *crisis characterization* and *treatment system*. As for *OntoProcessDeduction*, it aims at deducing the execution order of the selected services. It is based on treatment system an collaborative process.

OntoServiceState

The deduction of all the available services for the crisis response is based on key concepts which are (i) service (ii) risk (iii) consequence and the links between instances of its ontology. These links can be of two types: (i) *SAMEAS* which is an equivalent link (the instances are equivalent but named differently) (ii) *NEAR* which is a link of proximity (instances are not equivalent but close). At the moment, these links are created manually, future work consists in defining deduction rules in order to create them automatically.

According to the crisis, the state of actor services can take three values: (i) possible (ii) forbidden (iii) no opinion. This value depends on the links between instances. *i.e.* if the instance of service is forbidden by an instance of the crisis, this state is forbidden. If the instance of service reduces a consequence or prevents a risk of the crisis (or one of the instance linked with an instance of the crisis) and this state is not forbidden so the state of service is possible. However, an available service is more relevant if it prevents a risk of the crisis than a risk *NEAR*. Besides, in addition to the state, there is an indicator of confidence. The value of the indicator decreases with the number of *NEAR* links.

The deduction rules of this ontology were written in SWRL (Semantic Web Rule Language (O'Connor et al., 2005)) and executed by an inferrence engine.

For all entities belonging to the crisis, and for al equivalent entities, any forbidden service of that entity is a forbidden service of the crisis:

Crisis(?x) ∧ has(?x, ?y) ∧ Service_of_Actor(?z) ∧ isforbiddenby(?z, ?b) ∧ same_as(?y, ?b) → Service_isforbiddenby(?x,?z)

The following rule is used to select all services which can prevent a risk of the crisis:

Crisis(?x) ∧ has(?x, ?y) ∧ Service_of_Actor(?z) ∧ prevents(?z, ?b) ∧ same_as(?y, ?b) → Service_prevents(?z, ?y)

The following rule is used to select all services which can reduce a consequence of the crisis:

Crisis(?x) ∧ has(?x, ?y) ∧ Service_of_Actor(?z) ∧ reduce(?z, ?b) ∧ same_as(?y, ?b) → Service_reduce(?z, ?y)

OntoProcessDeduction

Collaborative process deduction is based on all the services selected by users and the priority order of treatment. Deduction rules are used to create tasks, gateways, events of the collaborative process and the order of execution of the tasks.

The following rule is used to create a task of the collaborative process from the real service of partner:

Service(?x) ∧ state_service(?x,"OK") ∧ Service_use_for(?x,?a) ∧ priority(?a,?b) ∧ swrlx:makeOWLThing(?y,?x) → Partner_task(?y) ∧ priority(?y, ?b) ∧ matchWith(?y, ?x)

The following rule is used to create task of the MIS, as well as message flows between the MIS task and the partners task:

Partner_task(?x) ∧ priority(?x,?b) ∧ swrlx:makeOWLThing(?y,?x) ∧ swrlx:makeOWLThing(?z,?x) ∧ swrlx:makeOWLThing(?a,?x) → SIM_task(?y) ∧ Message_flow(?a) ∧ mf_IN(?a,?y) ∧ mf_OUT(?a,?x) ∧ Message_flow(?z) ∧ mf_IN(?z,?x) ∧ mf_OUT(?z,?y) ∧ OrdreDeroulementProcess(?y,?b) ∧ messageflowWith(?x, ?y)

The following rule is used to define the execution order of tasks. We suppose that two tasks can be executed at the same time on condition that tasks are not provided by the same partner, do not reduce the same problem, and that the problem does not impact the same studied system component.

Service(?x) ∧ state_service(?x,"OK") ∧ Service_use_for(?x,?y) ∧ priority(?y,?z) ∧ Service(?a) ∧ state_service(?a,"OK") ∧ Service_use_for(?a,?b) ∧ priority(?b,?c) ∧ swrlb:greaterThan(?c,?z) ∧ impacts(?y,?d) ∧ num(?d,?f) ∧ impacts(?b,?e) ∧ num(?e,?g) ∧ swrlb:notEqual(?f,?g) ∧ provides(?h,?x) ∧ num(?h,?j) ∧ provides(?i,?a) ∧ num(?i,?k) ∧ swrlb:notEqual(?j,?k) ∧ Partner_task(?l) ∧ matchWith(?l,?a) ∧ Partner_task(?m) ∧ matchWith(?m,?x) ∧ SIM_task(?n) ∧ SIM_task(?o) ∧ Message_flow(?p) ∧ mf_IN(?p,?n) ∧ mf_OUT(?p,?l) ∧ Message_flow(?q) ∧ mf_IN(?q,?o) ∧ mf_OUT(?q,?m) → sameTime(?a,?x) ∧ sameTime(?n,?o) ∧ sameTime(?x,?a) ∧ sameTime(?o,?n)

SCENARIO

To illustrate collaborative process deduction in this section, a scenario was taken from a specific case of study: a NRBC exercise (27th of February 2004) managed by the *Prefecture du Tarn*[2] in France.

This scenario starts with the description of the crisis situation. Then, the application of the methodology will be presented step by step and graphics result will be shown.

Crisis description

The scenario played is the following: "At 10 AM, on 27th February 2004, the police is informed of an accident between a tanker truck (unknown substance) and a wagon containing chemical products (materializing a cloud). The policemen who were sent and the employees of the railway station fall unconscious while several children of the neighboring kindergarten (who were playing outside when the accident happened) feel sick.".

IsyCrisisTool: NRBC Crisis model

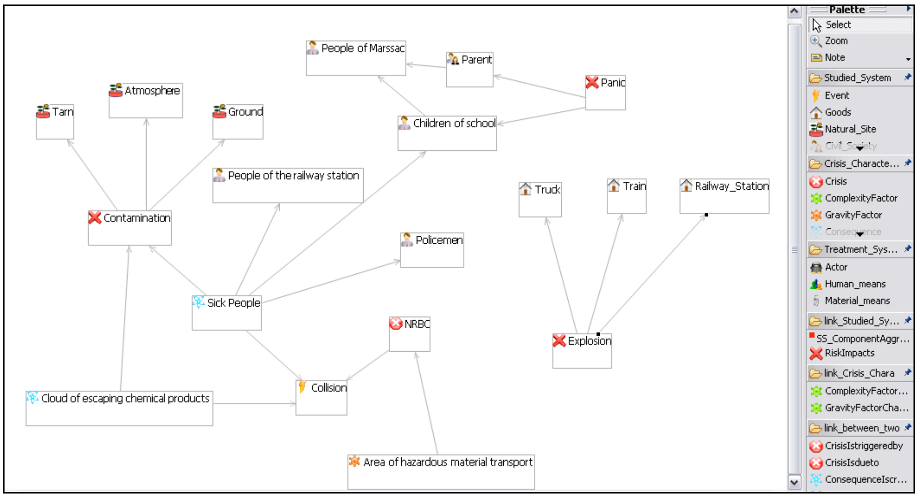

Figure 4. Example of NRBC crisis model

---

[2] *Prefecture du Tarn* is a French institution in charge of representing the government authority on a local scale (there are about one hundred *prefectures* in France).

The NRBC crisis model, which is shown in figure 4, is composed of the event of *collision* which triggers the crisis, the risk of *explosion* as well as the consequences of: *cloud of escaping chemical products* and *sick people* (which impact the *people of the railway station*, *policemen* and *children of the school*). The *cloud of escaping chemical products* creates the risk of *contamination* which can impact the natural sites. Moreover, there is a risk of *panic* since the *collision* happened near the school, some parents might like to take their children off the school.

***IsyServiceTool*: Partner service definition model**

In this example, we limited ourselves to four partners and some of their services. The whole definition services data was modeled through the *IsyServiceTool*.

***OntoServiceState:* available service deduction**

Once models have been injected into the ontology and the links SAME AS and NEAR have been defined between instances, the deduction rules can be executed. Before to give the result of the deduction rules , we explain one of them, the deduction of a service which prevent the risk of explosion. The deduction rules is the following : *Crisis(?x) ∧ has(?x, ?y) ∧ Service_of_Actor(?z) ∧ prevent(?z, ?b) ∧ same_as(?y, ?b) → Service_prevent(?z, ?y)*

?x is an instance of crisis, in our case *?x* is the NRBC crisis. Because of *has(?x,?y)* all the instances which were created in the model are save into the *?y* variable. Then all the service of actor which are defined in the *OntoServiceState* are saved into the *?z* variable. Likewise the *?y* variable, all things which are prevented by *?z* are saved in *?b* variable. Finally there is a comparison between the values of *?y* and *?b* and if these values are linked by a same as relation so the link Service_prevent between *?z* and *?y* is deduced. So service *?z* is an available service.

The result of the deduction gives us the following available services:

- Take care of people
- Prevent explosion
- Give first aid
- Set security perimeter
- Set medical post
- Injured person diagnosis
- Injured person evacuation

***IsyServiceState*: use service choice**

In this scenario, there are eight risks and consequences. The user needs to define the priority of the treatment of problems (one problem can be a risk or a consequence). Two "buttons" for each problem, allowing us to switch the lines of table and to modify the priority of the treatment of the problems as the priority corresponds to the number of the lines of table.

In this scenario, the priority of the treatment is as follows:

- Prevent risk of contamination
- Reduce Sick people
- Prevent Panic
- Prevent risk of not enough resource to take care of people.

Although other problems exist, it has been decided that there would be no other particular measure to avoid these risks, since these problems have less priority.

Service of actor	confiance	accesiblité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	actor	choix
SecurityPerimeter	1	1	???	1	???	policemen	◉ advisable ○ no_opinion ○ inadvisable
Service_can	1	1	???	1	???		○ advisable ◉ no_opinion ○ inadvisable

Service of actor	confiance	accesiblité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	actor	choix
Service_could	0,6	1	???	1	???		○ advisable ◉ no_opinion ○ inadvisable

Figure 5. Selection of services to solve the contamination problem

Then for each problem the user must select the services which will be used in the crisis response while starting with the problem having the biggest priority. Figure 5 shows the selection of one or more services to solve the problem *contamination*.

The user chooses the set of services to use to solve the problem. However the services are divided into two tables: the first *can be done* includes the services which solve this problem or a *SameAs* problem. The second table *could be done* includes the services which solve a *Near* problem. The closer the problem solved by the service is to the crisis problem higher the confidence indicator is.

Moreover, when a service is selected to solve a problem, if it also solves another crisis problem, the user must see the new state of all the problem solved by the service. In the example, if the user chooses to select the service SetSecurityPerimeter, the state of problems contamination, panic and explosion has been modified, as shown below.

At the end of this step, the following services are chosen:

- SetSecurityPerimeter
- SetMedicalPost
- CarePeople
- Prevent explosion.

Probleme	Commentaire	Etat	
Contamination		KO	+ -
Sick_People		KO	+ -
Panic		KO	+ -
not_enought_resource		KO	+ -
Cloud_of_escaping_chemical_products		KO	+ -
Fire		KO	+ -
Explosion		KO	+ -
Rail_Road_Accident		KO	+ -

Figure 6. State of problems before selection

Probleme	Commentaire	Etat	
Contamination		okcan	+ -
Sick_People		ko	+ -
Panic		okcan	+ -
not_enought_resource		ko	+ -
Cloud_of_escaping_chemical_products		ko	+ -
Fire		ko	+ -
Explosion		okcould	+ -
Rail_Road_Accident		ko	+ -

Figure 7. State of problems after selection

***OntoProcessDeduction*: collaborative process deduction**

The selected services are injected in the ontology. After that, deduction rules are executed. The result of this deduction is the execution order of services described below.

Service	Execution order
SetPerimeterSecurity	1
SetMedicalPost	1
Care people	2
Prevent explosion	3

Table 1. Execution order of services

The result shows that some services can start at the same time. We can explain it by the fact that each service is provided by different partners, solves different problems which impacts different components of the studied system. Moreover, the deduced order is not logical. Indeed, although problems of explosion and sick people impact the same component of the studied system, it is advised to execute these services at the same time. Finally, the deduced order needs to be validated. The validation is possible thanks to *IsyProcessTool*.

***IsyProcessTool*: collaborative process model**

The collaborative process can be validated by the user thanks to *IsyProcessTool*. At the end of this step, the collaborative process is transferred to a BPMN tool thanks to the model morphism explained in (Rajsiri et al, 2008). Figure 8 shows the final result.

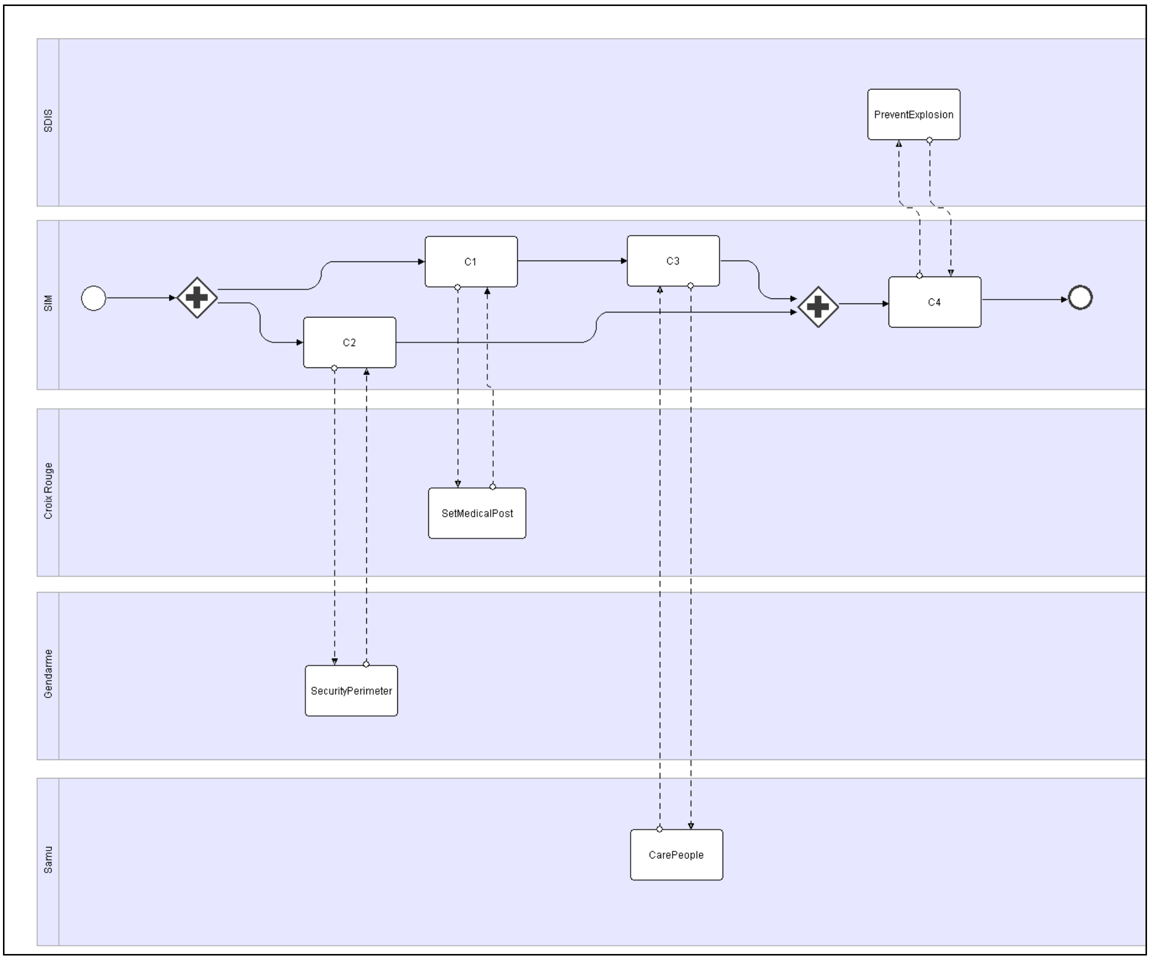

Figure 8. Example of NRBC crisis model

CONCLUSION AND FURTURE WORK

This paper presents the first step of a Mediation Information System Engineering. This step consists in deducing a collaborative process from crisis characterization and service definition models. This step is based on ontological deductions and crisis metamodel.

One of the future work consists in deducing automatically the links *SAMEAS* and *NEAR*. For the moment, these links are manually defined into the ontology. The next objective is to determine these links thanks to deduction rules. These rules will be based on attributes which needed to be define for each concept.

An another future work consists in bringing agility in MISE approach. Indeed, this approach needs to be agile because of the evolution of the crisis or of the crisis response. More especially, this approach needs to be agile at the tree level, which are business, logical, technological.

REFERENCES

1. Bénaben, F., Pignon, J.P., Hanachi, C., Lorré, J.P. and Chapurlat, V. (2007): Interopérabilité des systèmes en situation de crise, WISG'07, ANR & DGA Interdisciplinary Workshop on Global Security, Troyes, France.
2. Truptil, S., Bénaben, F., Couget, P., Lauras, M., Chapurlat V., and Pingaud, H.(2008): Interoperability of Information Systems in Crisis Management: Crisis Modeling and Metamodeling. INTEROP-ESA'08, Springer, Berlin, Germany.
3. Bézivin, J.(2004) "Sur les principes de base de l'ingénierie des modèles", RSTI – l'objet. Où en sont les objets ?, pp145--156
4. Luzeaux, D, Ruault, JR (2008) Système de systèmes, concepts et illustrations pratiques, Lavoisier, Paris, France, 36
5. Luzeaux, D, Ruault, JR (2008) Système de systèmes, méthodes et outils, Lavoisier, Paris, France, pp151--154
6. O'Connor, M., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., Musen, M. (2005).: Supporting rule system interoperability on the Semantic Web with SWRL. In: Proceedings of ISWC'05, Galway, Ireland
7. Maier, M. W., (1998). Architecting principles for systems-of-systems, Systems Engineering, Volume 1, Issue 4, Wiley InterScience, pp.267--284.
8. Gruber, T-R (1993) A translation approach to portable ontologies, Knowledge Acquisition, pp 199--220.
9. Sage, A.P., Cuppan, C.D. (2001): "On the Systems Engineering and Management of Systems of Systems and Federations of Systems. Information Knowledge, and Systems Management", IOS Press, Amsterdam, the Netherlands.
10. Touzi, J, Bénaben, F, Pingaud, H (2008): Model Transformation of Collaborative Business Process into Mediation Information System. IFAC'08, Elsevier, Seoul, Korea
11. Rajsiri, V., Lorré, J.P., Bénaben, F., Pingaud, H (2008): Collaborative process definition using an ontology-based approach. Proceedings of Pro-Ve'08, Springer, IFIP, vol.283,pp205--212. Poenan, Poland
12. Konstantas D., Bourrières J.P., Léonard M. and Boudjlida N. (2005): Preface of Interoperability of enterprise software and applications. proceedings of INTEROP-ESA'05, Geneva. Springer-Verlag; p. v-vi.
13. GMF: http://www.eclipse.org/modeling/gmf/
14. Protégé: http://protege.stanford.edu/