An Anticipative Effects-Driven Approach for collaborative process verification

Nicolas Daclin, Vincent Chapurlat, Frederick Benaben, Jean-Paul Pignon, Yves Guilloux

Internal Research Report / Rapport Interne de Recherche n° RR/11-04

An Anticipative Effects-Driven Approach for collaborative process verification

N.Daclin, V.Chapurlat, F.Benaben, J.-P.Pignon, Y.Guilloux

Ecole Nationale Supérieure Mines Alès -- 6 avenue de Clavière, 30319 Alès cedex
LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production, Parc scientifique Georges Besse, F30035 Nîmes

An Anticipative Effects-Driven Approach for collaborative process verification

N.Daclin[*+], V.Chapurlat*, F.Benaben**, J.-P.Pignon***, Y.Guilloux***

** LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production, Site EERIE de l'Ecole des Mines d'Alès, Parc Scientifique George Besse, 30035 Nîmes cedex 1 – Email : surname.name@mines-ales.fr*
*** DRGI – Ecole des Mines Albi-Carmaux Ecole des Mines d'Albi-Carmaux, Campus Jarlard, Route de Teillet, 81000 Albi, France - Email : benaben@enstimac.fr*
**** Thalès Communication, Colombes, France – Email: surname.name@fr.thalesgroup.com*

***Abstract**: the aim of this paper is to describe and illustrate a collaborative process model verification approach. This approach allows the coherence of a given collaborative process involving numerous resources, activities, and flows to be analyzed in a mission for which there are common objectives to achieve. It also enables the potential effects of this process to be to detected, characterized, and formalized so as to identify the effects on the mission to be executed, on the length of the process and its effectiveness prior to the execution phase. This approach is based on several principles formulated by systemic, enterprise modeling, and model-based system engineering which are formalized. The corresponding framework and support tool is then presented. Finally, the overall approach is illustrated on the basis of a crisis management process.*

***Keywords**: Collaborative process modeling, Collaborative process engineering, Verification, Validation, Property, Formal approach*

1. Introduction

The actors involved in various collaborative processes aim to design a product or a service, to manage production operations, and to support crisis management systems ((Couget et al. 2007), *etc.*). These actors come from different organizations, and must interact efficiently with others based on trust. In this process, they have to share data, knowledge, best practices, resources, and skills with confidence. However, at same time, they have to be sure of the relevance and quality of their roles and actions throughout the collaborative process, in order to achieve the expected results (*i.e.*, the desired effects of the process). Typically, their actions have to first be coordinated and synchronized. Second, even if the main desired outcomes of these actions are achieved, some others results (unpredictable and sometimes undesirable) due to interactions, how the environment changes, and other causes may be induced and lead to a worsening of the situation.
The goal of this paper is to present and illustrate an Anticipative Effects-Driven Approach (AEDA) allowing the potential effects of a given collaborative process to be detected, prior to the execution phase, in order to make this process more robust. This method is based on a formal verification approach applied to collaborative process models. It aims to achieve two objectives: on the one hand, to verify the overall consistency of the process (capacity, aptitude of actors involved in the common mission, fulfillment of objectives, and triggering condition of activities); on the other hand, to characterize and describe some potential (direct or indirect) effects of the collaborative process on its environment and resources. To perform this second step, the effects that may be dreaded in these processes must be (1) formalize and classified, and (2) mechanisms must be acquired for detecting, qualifying, and quantifying them. Thus, the approach we propose deals with principles and proof of property mechanisms that translate consistency requirements and allow expected or dreaded effects to be characterized.
This paper is organized as follow. After this brief introduction, the issues addressed and needs are described in the second section. Section 3 presents a state of the art concerning the different concepts used to set up the proposed AEDA approach. The fourth section introduces a thread example used to illustrate the various concepts in the approach and their relationships. The final section presents the conclusion, and the future perspectives for this research.

2. Problematic and needs

Models (*i.e.*, abstractions of the real world) are built and used throughout all the life cycles of systems, from their engineering (often re-engineering) to their dismantling, and including their exploitation. According to the hypothesis of Model-Based Systems Engineering (MBSE) (Estefan, 2007), such models are essential to better understanding and communicating facts to others, analyzing behavior, arguing, and evaluating alternatives related

+ Corresponding author Nicolas.Daclin@mines-ales.fr Tel. (+33) 466 387 039

to the system under study. This paper will initially focus on the engineering phase of collaborative processes. A collaborative process involves several partners, from different businesses or related businesses, with their resources and their own objectives to achieve. These partners have to work together on short-, medium-, or long term-based collaborations while continuing their own mission, within a given finality. In fact, a collaborative process has a strategic significance for the partners involved. For instance, a company can be involved in a product design process and take part in various collaborative activities that are reliable and economically interesting, but where there are still risks involved.
In the same manner, a common objective can induce some disquiet and side effects that are difficult to assess at the beginning of the collaboration. For instance Figure 1 illustrates this problematic within a crisis management process, in which various actors have to perform activities with some potential effects on how the crisis will evolve, on the environment, and also on the other partner's tasks.

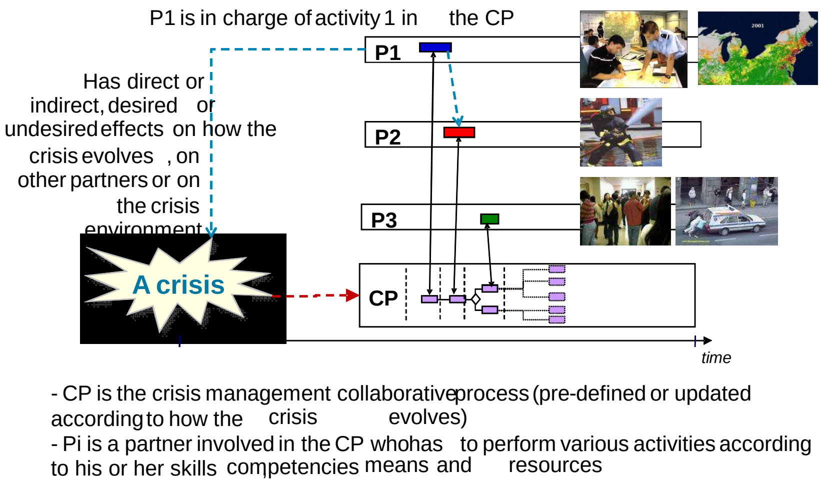


Figure 1. Illustration of a crisis management collaborative process

Each partner has to be able to analyze his or her role, position, behavior, and the possible risks that can occur throughout the life cycle process and before the effectiveness of the collaboration. In other words, the effects (expected or not) brought about by their very involvement in the process considered can not only be associated with performance objectives in terms of costs, duration, and quality of service/product (*i.e.,* adequacy between product/service, partners' needs and customer demands). The understanding, definition, robustness when faced with undesirable events, management, and control of collaborative processes have become crucial issues in a globalized environment, characterized by an impermanent market and fierce competition.
In this way, numerous initiatives (INTEROP, 2003; ATHENA, 2003) developed over the past years, and have to do with interaction between partners, have shown that partners' ability to be interoperable is a key factor for the success of their partnership. Moreover, long considered to be only an issue for computer science (IEEE, 1990), these initiatives have demonstrated that the concept of enterprise interoperability is now considered to be crucial (ECR, 2008), and is relevant for developing research area (EIRR, 2006), and can take place at different levels in an organization (Chen *et al.*, 2007) (*e.g.* business, process, or service). In the limited framework of collaborative processes, partners are interoperable if they are able to share data, services, skills, resources in order to fulfill a common mission. Thus developing their interoperability abilities and capacities allows them to ensure the success of the whole process, and therefore to ensure a given level of performance, efficiency, reactivity, and agility in this process. However, this process can be affected by a lack of interoperability between partners during the process runtime. Likewise, allowing them to know the potential effects of their collaboration prior to the execution phase could help (1) partners to anticipate poorer than expected results in terms of achieving the process objectives, and thus (2) to adapt the process as much as possible. In this case it is about to analyze each effect that a partner can induce on the process.
We see that collaborative process engineering is based on reflection, modeling, and analysis. Various modeling languages and tools can be used to model expectations. For the purposes of analysis, some of these tools can provide - in some cases and under various hypotheses - verification mechanisms (*e.g.,* checking model coherence), validation mechanisms (*e.g.,* simulation or model expertise regarding the common mission), and evaluation and optimization rules used before the process is executed. All of these activities help to increase partners' confidence in each other, highlight potential deficiencies, dispel doubts, and find potential improvements. Thus, these tools make techniques available that are more or less formal, from expertise to simulation, less frequently to formal techniques.

The AEDA approach has to provide:

- A set for modeling means (*i.e.*, a set of concepts, relations, and rules for modeling the process, the different configurations and characteristics of any actors involved, the potential effects, and the environment in which this collaborative process takes place.
- A set of reasoning mechanisms for checking the possible effects induced by each action that partners may execute during the process.
- A set of improvement rules for guiding partners so as to improve their interoperability and reduce the causes of the effects detected. This set is not described in detail in this article.

3. State of the art and discussion

Detecting and analyzing the effects that can be generated by a collaborative process presupposes that a model of this process exists. A process model provides a representation of the different actions, resources, controls, and other knowledge, such as the objectives and mission, about how actors are involved in a partnership. In addition, it allows its effects to be highlighted and characterized so as to ensure that it can be improved. Most approaches dedicated to the study of processes are based on modeling techniques used to analyze and design systems (*e.g.*, ICT systems, and industrial systems). The first research on processes was undertaken at the end of the 1970's in the form of SADT (Structured Analysis and Design Techniques (Ross, 1977)). They highlight the necessity of using graphic language to build and validate systems. Other similar approaches were then developed in the 1980's, such as the IDEF – standing for the Integrated Computer Aided Manufacturing DEFinition method (ICAM, 1981). More recent approaches, such as the BPMN (Business Process Modeling Language, mainly deployed in the ICT field, allow all business users to (1) model a process, (2) implement technology that perform a process, and (3) manage and monitor a process (OMG, 2009).

In a similar domain, related to the sharing and the understanding of process models and process information, methods such as Process Specification Language (PSL) (ISO 18629, 2004) and POP* (Process Organization Product) (ATHENA, 2005) have been developed. The need to develop this kind of approach stems from the globalized environment in which enterprises are doing business today. Indeed, they must increasingly share their enterprise models to organize, manage their relationships, and ultimately, make successful partnerships. PSL is an interchange format designed to help exchange process information automatically across a wide range of manufacturing applications, such as process definition, process planning and scheduling, and simulation tools. POP*is a neutral meta-model for mapping (both semantic and syntactic) between different modeling languages. POP* addresses mainly process model mapping, but it can also be used for other mapping, such as data model mapping.

All these approaches are dedicated to the development of processes, either with the intention of modeling a given process in a given context, or with the intention of improving the understanding and sharing of information embedded in a process. Basically, process modeling languages are used to model a sequence of activities from the beginning to the end. Depending on the language, this will include more or less concepts (*e.g.*, resources and control in SADT, message flows, pools in BPMN). However, they do not fully consider attributes that characterize and define the elements involved in a process. As a consequence, it is necessary to allow actors to gather a maximum amount of knowledge about elements. For instance, the time space and shape (TSS) frame of reference (Le Moigne, 1977) can allow this knowledge to be collected for any element involved in the process. Indeed, by adapting this frame, it is possible to position an activity, resource, control, or any other element in this frame (TSS). Moreover, this knowledge, represented in the form of attributes that characterize the element considered, can be independent of any domain of application (*e.g.*, resource capacity, availability, pre-emption (Vernadat, 1996)), or specific to a given domain (*e.g.*, an activity can require a certain level of protection for its resources in a crisis context). It is on the basis of this enrichment, in terms of attributes, that reasoning about the possible effects can be achieved.

The anticipative effects-driven approach we propose is complementary to the other analysis approaches. Its main interest lies in its capacity to model, analyze and assess, and characterize the effects that occur in a collaborative process. Thus, in terms of approaches that focus on the assessment of the effects that a given system can produce before its implementation, Failure Mode and Effects Analysis (IEC 60812, 2006) is certainly the most widespread and commonly used method. Based on its success, some extensions to this method have been developed since its creation. In this way, the Process-FMEA (also named P-FMEA) helps to identify the effects that a process can have on a product. This approach establishes a set of potential failure modes and effects of failure that can occur throughout the process runtime in order to take corrective action - before the process implementation- and to eliminate potential failures. P-FMEA is commonly used in manufacturing processes, and is based on a brainstorming procedure applied every time a process has to be implemented.

Other approaches, such as the Effects-Based Operation theory (Lowe et al., 2004; Smith, 2002) consist in characterizing and evaluating the potential effects of actions, which are supposed to lead to a final outcome. Although the EBO construct concerns the characterization of effects, contrary to FMEA, it is not based on the search for the potential effects that can be generated by actions involved in a process. It focuses on the search for outcomes that actions have to achieve in a given process. EBO theory offers (Batschelet, 2002) a complete methodology from the knowledge phase to the assessment phase, *via* a planning and an execution phase, to observe and analyze the effects resulting from the execution of a process. It is worth noting that the effects based operation approach comes from and was developed in a military context.

Both approaches presented above, are interested in determining the effects that can be produced by actions involved in a process. Despite the fact that they address this problem from two different points of view – P-FMEA establishes potential effects before execution, and EBO analyzes effects right after execution – their common goal is to improve processes involving many activities. Nevertheless, these approaches lack certain elements, which are taken into consideration by our anticipative effects-driven approach. First, no formal methods are proposed and implemented to detect effects (*e.g.*, P-FMEA is based on a brainstorming procedure, EBO on evaluation by means of an assessment phase). This deficiency can lead to the oversight of potential effects, and a poor evaluation of them. In this case, the interest of formal techniques, such as verification techniques (Dindeleux *et al.*, 1998; Dubois *et al.*, 1994), is (1) to provide proof about possible changes in the process, independently of any human interpretation, and (2) to focus actors' attention on a given phenomenon or highlight a crucial situation, which has not been taken into consideration.

Furthermore, none of these systems proposes a precise evaluation of the nature of effects (severity is nevertheless considered in P-FMEA for instance). Yet, it is important to characterize effects accurately (*i.e.*, to propose a characterization method that allows actors to know what are really the potential impacts on the collaborative process under study).

4. Example thread: a collaborative process for crisis management

The approach we propose is illustrated on the basis of a collaborative process for crisis management. However, the approach can be applied in others contexts in which collaborative processes must be managed and analyzed (*e.g.*, interoperable industrial systems) (Mallek *et al.*, 2011). This process has been implemented in the French research project ISYCRI (Interoperability of SYstems in CRIsis situation, ANR-06-CSOSG). The scenario is related to an accident involving a freight truck that is probably transporting hazardous substance (the nature of substance cannot be clearly identified at the beginning of the scenario for multiple reasons). The accident occurs on a railway junction, near a medium-size town. The truck may explode and this would have a major impact on the population, the passengers in train that has stopped just before the junction, and the natural environment. The process to manage this kind of accident is known and has already been described by a set of procedures and an intervention process. It is controlled by the head of public services and involves resources such as the office of infrastructure, the police force, the emergency ambulance service, and firefighters.

Before presenting anticipative effects-driven approach concepts, the following must be defined and taken into consideration:

- The initial collaborative intervention process as it has been defined and proposed by the emergency planning unit. BPMN was the modeling language chosen for this purpose. The next sections show why it is necessary to perform some BPMN concepts and relations enrichment to bring out, further, the effects generated by the collaborative process.
- The requirements that are imposed by the emergency planning unit. These requirements are related to the duration of the process, expected skills, resources, and other such concerns.

The following figure illustrates the collaborative process which describes the crisis scenario management. Activities are performed both in sequence and in parallel. Activities use resources (not shown directly in this model) and each element is characterized by a set of TSS (Time, Shape and Space) attributes that are defined by actors. For instance, an activity embeds attributes, such as task name, definition of purpose, person in charge, authority, definition of mission, mission horizon, mission period, mission type, required aptitudes, and resources used.

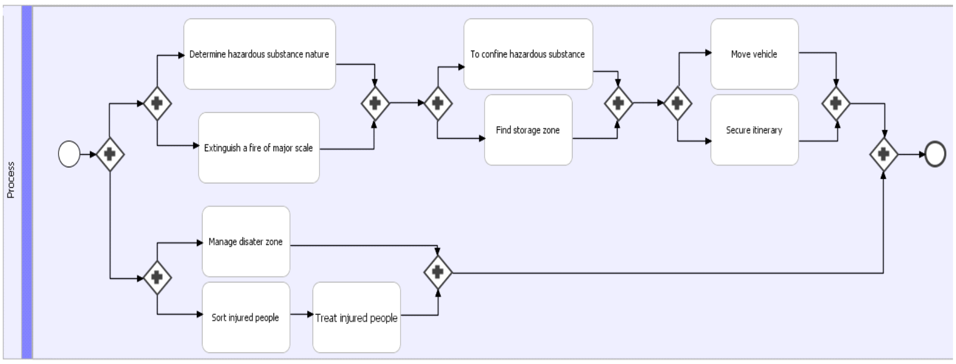


Figure 2. Collaborative process proposed for crisis management

5. AEDA

The following sections set up the conceptual principles of our approach, and illustrate them on the basis of the example given in Figure 2. An effect first has to be characterized and modeled. Then, in order to take into consideration the proposed effect model, it is necessary to enrich the process modeling language chosen (BPMN), at a conceptual level. Finally, it is necessary to check:
(1) If the modeling requirements are respected (*i.e.*, to verify the compliance of the model with its meta-model and modeling guidelines. This step aims to verify model coherence and quality.
(2) If the requirements coming from the collaborative process are respected. This step aims to partially validate the model by matching it with the perceived reality of the collaborative process. This requires then, the support of domain experts.
(3) If some of the feared effects, difficult to predict and potentially harmful, are not to dread.
The methodological steps of the AEDA approach are then presented.

5.1. *AEDA modeling principles*

Effects characterization

An effect is first characterized by its nature (*i.e.*, what is its potential scale of significance?) and its structure (*i.e.*, what are the *objects* concerned by this effect? What are the relations to consider between them?).
An effect is defined as a *situation* that can be expected, undesired, or dreaded, which results from an *interaction* between one *object*, considered to be the source of the effect, and one or several *objects*, considered to be the destination of the effect, as illustrated in Figure 3.

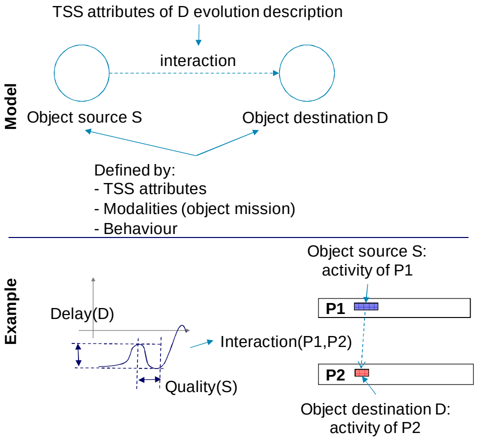


Figure 3. Effect model principles

Object models are physical or logical elements present throughout a process: activities, partner's resources and elements that compose the environment that can be affected by the process (Couget *et al.*, 2007). They provide a meta-model for crisis management systems, which describes relevant objects to be considered in this domain.
An object is specifically defined by a set of *Time, Shape and Space (TSS) attributes* and a set of *modalities* (Maier, 1995).

- A *TSS attribute* characterizes any element, from a quantitative or qualitative point of view, that changes over time (duration), spatially (location), or considers its shape (*e.g.* capacity). The TSS attributes list (summarized in Figure 4) allows the required attributes that have been selected for the crisis management process under study to be defined and formalized. Finally, any *object* may be "a part of" or "interacts" with another element. In this case, the way each *object* evolves affects and modifies the referential of the surrounding *object*. Thus, defining which *objects* evolve in a given referential also allows us to know the impact of these *objects* on their environment.
- The *modality* of an object describes the systemic nature and role of the object (*i.e.*, the classical expectations the object must respect when facing its environment and exchanging flows). It is composed of five expectations:
  - *To know*: the set of internal data, information, knowledge, events, skills, and abilities expected to allow the object to fulfill its mission.

- *To want*: the set of required inputs (skills, abilities, information, data, knowledge, events, materials, energy) needed to control the (set of) action(s) to be done by the object in order to fulfill its mission.
- *To have to*: the set of outputs that has to be provided by the object so that it fulfills its mission and achieves its purpose. It can be an output itself (competencies, skills, information, data, knowledge, events, materials, energy) used by other activities that have ‘to have to’ or ‘to want’ modalities.

date	- begin - end - involved in the crisis - out of the crisis
duration	- desired duration of action - desired date of action impact
influence	- recognition - authority - hierarchy
dimension	- volume - surface - height - length - width - mass - distribution - density
vulnerability	- human - environmental - in space - organizational
quantity	- quantity - size - number of same objects
complexity	- organic - control
cost	- acquisition costs - cost of use - cost of maintenance
location	- position (GPS, other)
dynamic	- other TSS attributes variation

Figure 4. TSS attributes list

- *To have*: the set of inputs (skills, abilities, information, data, knowledge, events, materials, energy) to be processed by the object (*i.e.*, allowing it to provide its outputs and achieve its mission).
- *To be able to*: the set of inputs (as resources) that can provide the skills, abilities, data, information, knowledge, materials, events and energy expected and allow the object to provide its added value and fulfill its mission. These can be also named ‘input resources’.

Figure 5 illustrates the modalities of the Activity object.

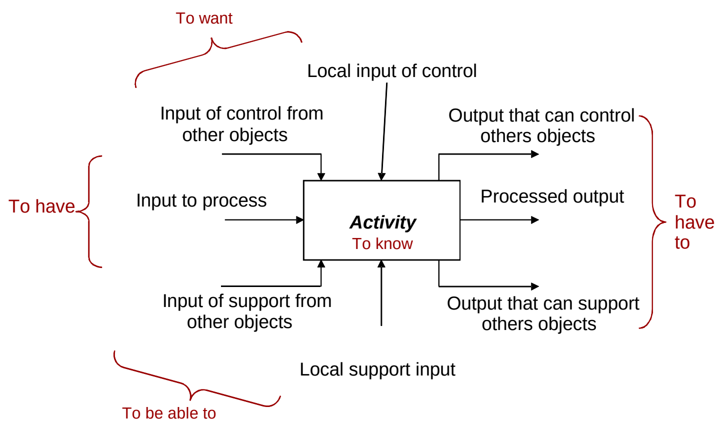


Figure 5. Activity modalities

This notion of modality allows the different *interactions* to be identified. The concept of interaction (Leger *et al.*, 1999) allows one to formalize how, in what conditions, and with what effects an element can dynamically interact with another one. Interactions are defined as:

– The “know-how” (KH) interaction represents the flow of knowledge and skills;

– The "want-do" (WD) interaction represents the flow of input that triggers the object;
– The "can-do" (CD) interaction represents the flow of inputs that are considered as resources;
– The "must-do" (MD) interaction represents the flow of final outputs.

All the interactions presented above are concretized by physical (material, information, energy) or logical flows (influence, authority…), which are necessary between the objects and can be the causes of potential effects. Thus, an *interaction* between two objects is modeled by a set of rules highlighting the resulting variation(s) of one (or more) Time, Space and Shape attribute(s) of the destination under the action of the source (*e.g.* activity A1 onto activity A2 may modify delay attributes of A2, resource R1 onto activity A3 may modify the quality of service of A3). An *interaction* is not meant to explain the resulting changes, but aims to concretize and formalize what these changes are. For instance, the following rules are expressed in natural language to make them easier to understand:

- A*n activity aiming to confine a dangerous product may induce the risk that this product will explode*.
- A *partner selected as a potential resource to perform an activity has to provide a capacity at least equal to the capacity requested to perform the considered activity, in order to avoid delays and maximize the efficiency of the activity*.

An effect can be defined as:

- Predictable (*i.e.*, assessable and indicators exist for the source object and the destination object(s).
- Potential, in this case a logical relationship exists between the cause and the effect.
- Unpredictable or emergent (*i.e.*, there are no indicators or logical relationships that allow the effect to be determined. The characterization of this kind of effect is not considered by our approach.

Furthermore, an effect can be:

- *Direct* (also named 1st order effect) if and only if it is directly extracted from an interaction between a source and a destination (*e.g.*, controls an activity).
- *Indirect* (also named 2nd/nth effect) if it is the result of a 1st order effect. For instance, a resource is not adapted to perform an activity A1, and this may induce delays (*i.e.*, harmful indirect effect on the next activity, A2).

Finally, an effect can be characterized by its nature. (Mann, 2002) defines four natures of an effect:

- *Harmful effects* are produced when the source induces a deterioration of the characteristics of the destination. These kinds of effects have to be broken down.
- *Good effects* are produced when the source induces a variation of the characteristics of the destination as expected. These kinds of effects have to be maintained.
- *Excessive effects* are produced when the source induces a variation in the characteristics of the destination beyond expectations. These kinds of effects have to be reduced.
- *Insufficient effects* are produced when the source induces a variation in the characteristics of the destination less than expected. These kinds of effects must be improved to become efficient.

Figure 6 shows the application of the rules proposed above on the crisis management process introduced in Figure 1. Three kinds of effects are brought to the attention of actors involved in the process. A harmful effect is detected in the activity "to confine a hazardous substance" requiring corrective actions to be deployed. A good effect is highlighted on the activity "to determine the nature of the hazardous substance" and an insufficient effect occurs on the activity "to treat injured people". In this case, appropriate actions have to be implemented to tackle this deficiency.

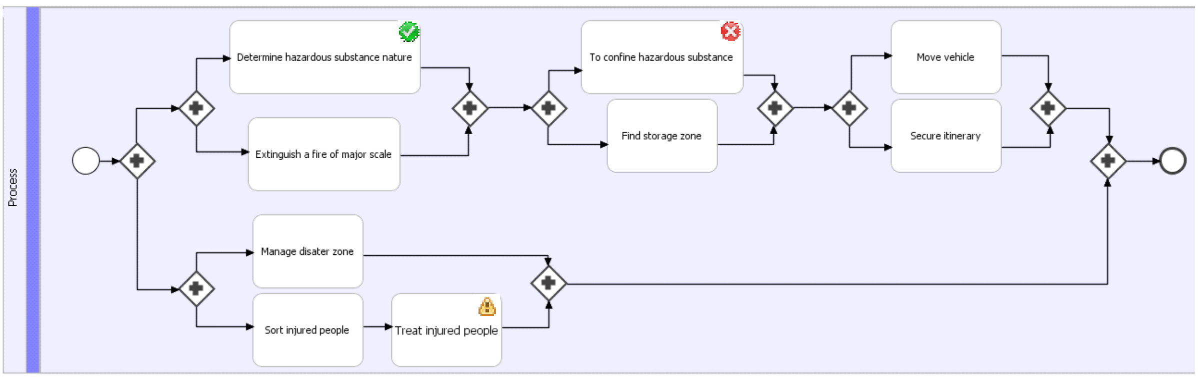


Figure 6. Illustration of the effects principle in the collaborative process for crisis management.

The nature and structure of an effect are not sufficient to highlight it on the process model. Indeed, it is difficult – or even impossible – to highlight an effect directly. It is also difficult to show a potential effect that is initiated by the combination of other effects. This demonstration requires one to provide a formal model of the effect in order really to verify it and to facilitate the implementation of propagation mechanisms allowing $2^{nd}/n^{st}$ order effects to be detected. Moreover, this modeling step allows formal techniques to be taken into consideration.

Effect formalization for proof: properties

In order to bring out the potential effects generated, it is necessary to apply these on the collaborative process submitted to analysis, which requires the previously defined effects model to be formalized. In this way, the modeling can be envisaged from two points of view. On the one hand, an effect can be modeled as a property to act as a proof. In this case, the goal is to verify formally on the process model the property that represents the effect. On the other hand, an effect can be modeled using a mechanism that ensures the propagation of a given effect throughout the process being modeled. This kind of model is mainly used to act as execution.

If the effect is concerned by a property, it is related to a causal temporized and constrained relation between two predicates called cause and conclusion (Lamine, 2001). Cause and conclusion are described using attributes (*i.e.*, information that characterize elements involved) or functions extracted from the collaborative process model. Indeed, this description allows one to provide an explicit expression of (1) the source that initiates the effect, (2) the destination that is affected by the source, and (3) the possible variation of both of these elements. Finally, the causal relation (*i.e.*, implication) describes the nature of the relationship between the cause and the conclusion and the temporal constraints on which this relationship is based. An effect characterized as being good and its property model is represented in Figure 7. This property can be applied to the collaborative process for crisis management in order to ascertain the effect of resources that are allocated to activities.

Effect	**Cause**	**Relation**	**Conclusion**
Good	Each partner, to be allocated to a given activity, has the required aptitude, is available, and is at the right location	implication	Partner is eligible to be allocated to the activity
	Exists *a* in Activities, forall *x* in Partner, [requiredAptitudes (*a*) in aptitudes(*x*) and location(*x*) = location(*a*) and availability(a) = true]	⇒ [for all t]	[x in eligibleresourceOf (a)]

Figure 7. Example of effect modeling based on a property

For example, the property defined as:

Forall a in Activities, Forall element in input(a) [requestTSSInput(a) in TSS(element)]

$\Rightarrow$

[effect(a, element):= good]

means that if the TSS attributes requested by a given activity as input contain the TSS attributes of the element considered to be processed, then the effect of the activity on the element is considered to be good, and this is true for all the activities in the collaborative process.

This property can be decomposed into sub properties that specify the way to interpret the variation of effect value, with the consideration of a more precise subset of TSS attributes of the activity and/or of the element.

Thus, the following property:

Forall a in Activities, forall aptitude in requestedAptitude(a), forall partner in elligibleResourcesOf (a)
[requestedCapacity(aptitude,a) < capacity (partner)]

$\Rightarrow$

[effect (partner, a):= insufficient and elligibleResourcesOf (a):= elligibleResourcesOf (a)-partner]

means that if an activity requires a precise capacity (evaluation *i.e.*, quantification or qualification of a given aptitude according to a common scale of measure), then each partner selected as a potential resource must provide this capacity in order to be qualified. Otherwise, the effect of the “t” element has a sub-attribute quantity less than the sub-attribute quantity that describes the expected quantity of elements for the activity (*i.e.*, the quantity of inputs from this partner for activity “a” is considered to be insufficient).

The modeling of an effect as a property provides a support of reasoning, allows the collaborative process to be analyzed and the accuracy of the characterization of an effect to be proven. However, it can appear that the characterization of an effect depends not only on the proof of one property, but on many. Moreover, an effect on a given element can also affect other elements in its environment, leading to its propagation throughout the collaborative process.

In this case, it is interesting to offer the possibility of implementing a mechanism allowing one to simulate and to model an effect and its impact according to its propagation. This can be done by using a properties tree as

introduced in (Chapurlat *et al.*, 2009). A property tree is a recursive mechanism based on three concepts: property, node, and relation.
The concept of property is related to the modeling of an effect as previously defined, while the concept of node represents an “abstract requirement” characterized by an effect as well. A characterization of an effect for an abstract requirement can depend on the characterization of (1) one or more effect by means of a property and/or (2), one or more effect from another node, characterized themselves by property(ies) or node(s). The concept of relation represents either logical operators or other functions that describe the conditions within an effect for a given node is reached. The consistent relationship between these three concepts allows a property tree to be obtained in order to characterize an effect based on the mechanism of propagation.
A basic example of a property tree is given in Figure 8. Applied to the collaborative process for crisis management, this tree enables a good effect related to the overall process organization to be highlighted and defined as: *someone is in charge of the process AND how activity is organized has a good effect*. The first term is a simple effect model using a property, and the second term represents the effect of an abstract requirement defined hereafter. Thus, the (good) effect of the abstract requirement activity organization is defined as: *any activity has a person in charge AND any activity uses resources AND how resources are organized has a good effect*.
Finally, the characterization of the effect of the abstract requirement resource organization is described as: *any human resources have responsible AND any material resources have responsible*.
Starting from the effects modeled by properties, the objective is to propagate their characterization throughout the entire structure, including the abstract requirements, in order to reach the final effect characterization. It is worth noting that the logical operator used for all abstract requirements is a logical “and”. In the same way, the logical operator “or” can be used depending on actors’ needs in term of the characterization of effects.

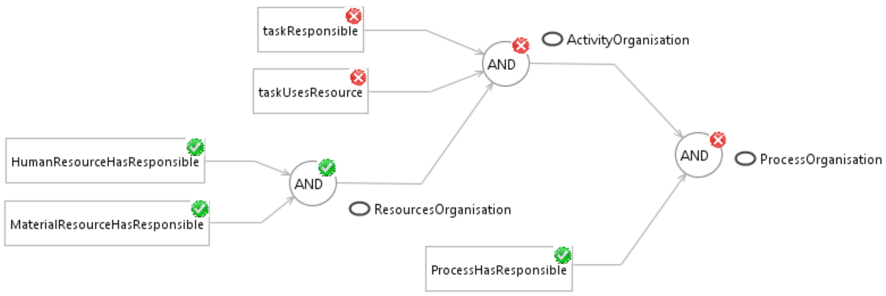


Figure 8. Example of using a properties tree to model an effect.

At this stage, it is important to remember that the success of the effects characterization depends greatly on the knowledge embedded in the collaborative process and given by the actors involved. As mentioned above, a maximum amount of knowledge is required to detect and refine a maximum number of existing effects. Thus, the characterization of an effect is closely related to the process modeling language used.

Conceptual enrichment of a process modeling language

As mentioned before, the characterization of an effect is closely related to the model used for the collaborative process and the information available (as far as possible) for its elements. Furthermore, the model for an effect considers and depends on the element attributes extracted from the process. In other words, an effect (characterization and modeling) has an impact on the modeling language itself. This means conceptual enrichment of the modeling language must be performed in order to capture a maximum amount of knowledge. Thus, depending on the context in which the approach is applied, this enrichment can be carried out in two ways (the semantic aspects of these enrichments are not presented here).
The first is related to the addition of various constructs to the modeling language. Although a given modeling language offers the possibility of reaching a full model for a specific field, it may require adaptations in order to be deployed in another field. This is the case for the BPMN language chosen, which is perfectly adapted to the Information and Technology context, but which lacks modeling concepts in other contexts (*e.g.*, example threads dealing with crisis management).
Thus, modeling a collaborative process for crisis management leads one to consider other constructs, such as human resources, materiel means, population, and civil society. As a consequence, these constructs have to be considered and added to the BPMN language. The following figure represents the implementation of two constructs implemented as modeling elements in BPMN (human resources and material resources).

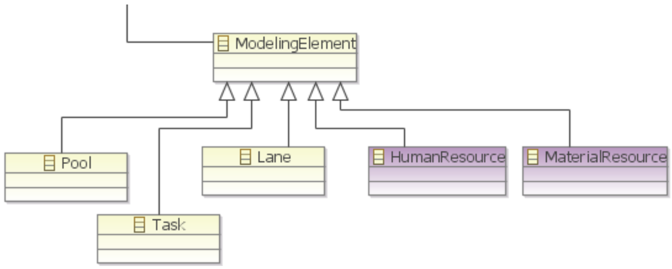


Figure 9. BPMN constructs added to the crisis management context

Once the first enrichment has been performed, the second consists in adding TSS attributes to modeling elements. In the same way, if the modeling language chosen allows - in its original version - to consider some attributes on its elements, it can be required to enrich these elements according to the context. To perform this kind of enrichment the Time, Space, and Shape (TSS) referential presented above is utilized. Figure 10 gives an example of a complete enrichment step for a task in a crisis context.

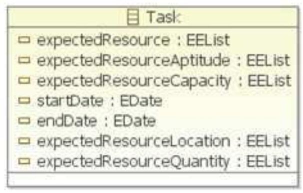
Modeling element	**Attributes enrichment**
	

Figure 10. Attributes enrichment for a task in a crisis management context

The result of these two steps is an "enriched BPMN" (partially presented here), allowing actors to model their collaborative process according to the context concerned, as well as to collect a maximum amount of knowledge about elements involved in its execution. The main AEDA conceptual enrichments related to effects, properties, and TSS attributes are summarized in the UML meta-model in Figure 11.

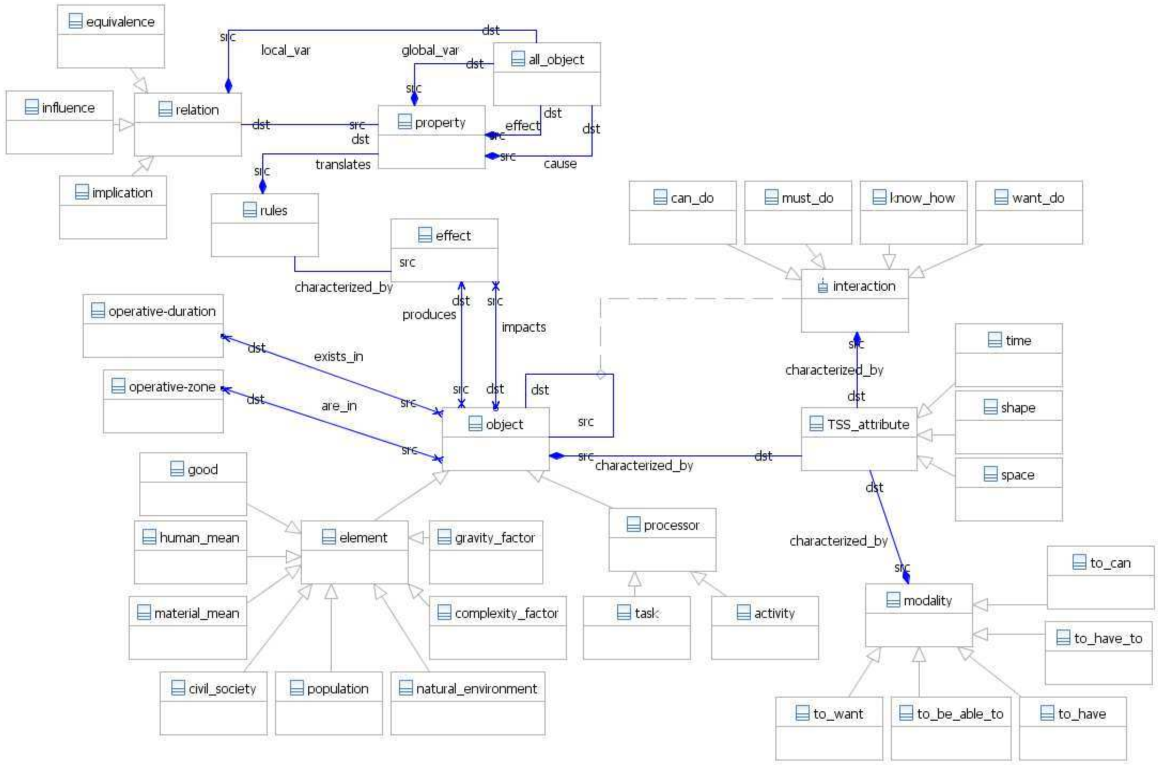


Figure 11. Main conceptual enrichments of AEDA (partial view)

5.2. *AEDA checking principles*

Effects characterization can be carried out in two ways. On the one hand, it can be performed by expertise. In this way, effects characterization is based exclusively on knowledge coming from an expert. This kind of technique cannot be formally proven and may be subject to reconsideration.

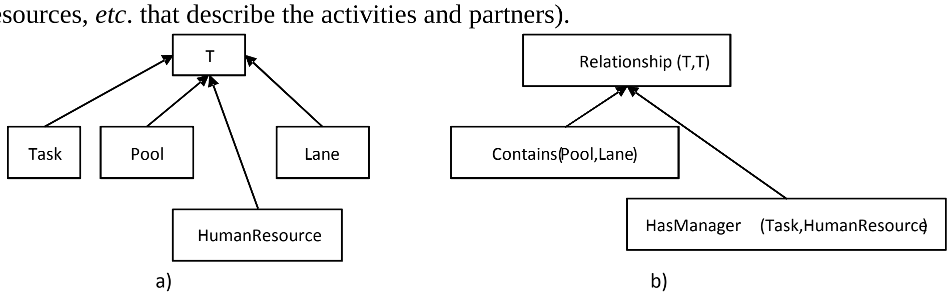
On the other hand, it can be performed using formal verification. In this case, it is necessary to implement verification techniques which allow the characterization of an effect, based on the knowledge extracted from the model, to be proven formally.

This paper focuses on the second technique, illustrated by a conceptual graph (Sowa, 1992; Chein *et al.*, 1992), which follows an approach similar to the one proposed in (Mallek *et al.* 2011). A conceptual graph allows one to reason *via* a graphical representation of the knowledge that is easy to understand and to manipulate. The technical implementation of the verification, using the conceptual graph, is performed with COGITANT (Genest, 2010). However, this implementation requires transformation in order to be applied and to characterize an effect. These transformations are performed with ATL (ATLAS Group, 20006), and concern enriched BPMN process model (1) transformation, (2) verification, and (3) effects analysis.

Process model transformation

Process model transformation aims to transform the collaborative process model into a Conceptual Graph according to the approach presented in (Mallek et al. 2011) and using two models called the *support model* and the *facts model*. This transformation is performed in order (1) to apply properties proof mechanisms to the process model and (2) to embed some operational semantics in the model itself in order to perform a propagation of the effects.

The *support model* allows a representation and formalization of all the concepts and relations from the enriched version of BMPN to be obtained. This support model schematized in Figure 12 is split up into a *concepts lattice* and a *relations lattice* by means of conceptual graphs theory. The concept lattice represents the hierarchy between concepts, and the relations lattice represents the relations between the concepts. Furthermore, this transformation also includes the individual markers which describe the instances of the concepts (*i.e.*, the activities, resources, *etc*. that describe the activities and partners).



Marker	**Concept**
Manage disaster	Task
PolicesForces	HumanResource

c)

Figure 12. Example of support model with concept lattice (a), relationship lattice (b), and individual marker (c)

The *facts model* is a conceptual graph, named also *graph model*, which considers concepts and relationship lattices and markers. It represents the model of the collaborative process translated into a conceptual graph. This model corresponds to the support model illustrated in Figure 13.

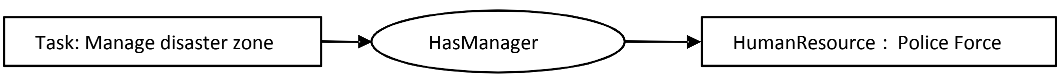


Figure 13. Example of a fact model in agreement with the support model

Verification is then performed on the *graph model* thanks to the mathematical mechanisms called *projection*. The projection step aims to formally establish a relationship between the *graph model* and the property model (translated, itself, into a particular conceptual graph named *constraint*), which formalizes (1) the modeling requirements to be respected, and (2) the effects as shown before.

The COGITANT tool is used for considering two kinds of projection, known as positive constraint and negative constraint:

- A positive constraint is composed of a cause and a conclusion. To verify a positive constraint, any projection from the constraint cause into the graph model must be able to be extended into a projection from the whole constraint graph into the graph model to consider.

- A negative constraint is a simple conceptual graph. If it is projected onto the graph model, then the constraint is not verified.

These constraints are then used for the two considered purposes as follows.

Modeling requirements verification

The following figure represents a negative constraint that allows model coherence verification to be performed. In this example, the property to verify is that *any task has an attribute name with the string value empty string*. If the graph is projected onto the graph model, the property is not verified, which means that some information is missing from the process model.

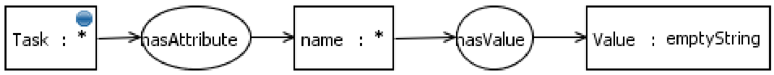


Figure 14. Example of a conceptual graph

Effects analysis

Effects analysis draws attention to all the potential effects that can be generated by the collaborative process, in order to validate them (or not). The effects model is transformed into a conceptual graph and verified. According to the result of the verification, the nature of the effect is displayed to the actors.

Thus, this verification (*i.e.*, effects analysis) results in the use of the projection mechanisms of the conceptual graph, which represents the model of the effect (also named constraint), onto the process model given by the support and fact files.

Figure 15 shows the conceptual graph that represents effects modeling: *An activity named Manage disaster, uses human resources named Police Force, and human resources named Firefighters, and human resources named Emergency Ambulance Service, and human resources named office of Infrastructure*. If the cause can be projected onto the model of the collaborative process, then the conclusion has to be projected too. Thus, if the projection takes place, the effect is characterized as being good.

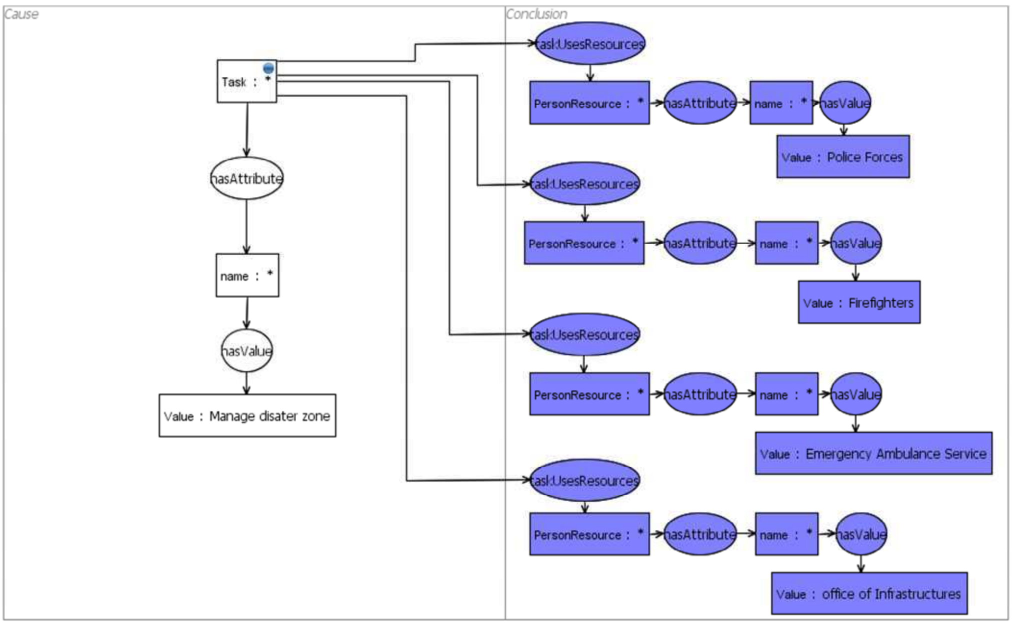


Figure 15. From effect modeling to the conceptual graph for effects analysis

Figure 16 shows and summarizes the previous concepts and relationships, which support the AEDA approach. The next part presents the different AEDA steps in use.

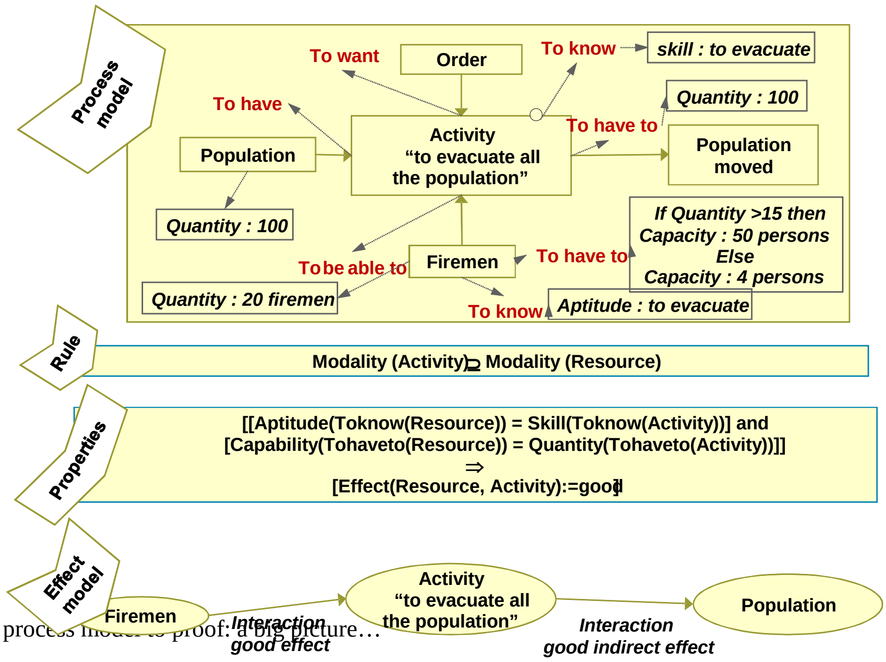


Figure 16. From process model to proof: a big picture…

5.3. *AEDA methodological steps*

The AEDA approach is implemented through different steps inspired from the Effects Based Operations (EBO) approach, and summarized in Figure 17 (Daclin *et al.*, 2009). These steps aim (1) to characterize the elements (activities, resources, data, and control flows between them) involved in the collaborative process (use of the TSS Frame of Reference), (2) to define the effects that can result from its execution (use of the Effects description rules base), and (3) to check the process model in order to detect if conditions under which effects can occur are verified.

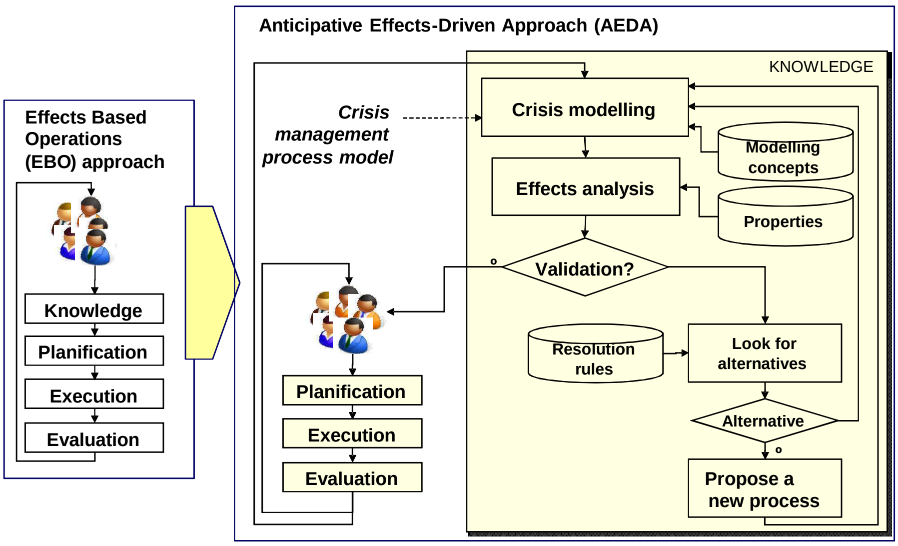


Figure 17. The AEDA placed in an EBO approach context

After building the aforementioned collaborative process model, the first step – *characterization of elements* – aims to define precisely all the elements involved in the process. In this way, the objective is to collect a maximum amount of information about the different activities, the resources that have to perform activities, and the controls, as well as expected input and output flows from activities. This step is supported by a TSS Frame of Reference which structures information about elements. The TSS Frame of Reference represents a set of attributes applicable to each element (*e.g.*, activity start time, resource name…). The purpose of this step is not to be exhaustive. It allows one to represent the basic knowledge of partners regarding the process to be

implemented. However, the more the characterization of specific elements is complete and precise, the more the next step is able to highlight the effects.
The second step – *characterization of effects* – is intended to explain to the partners the effects that may be induced by the collaborative process to implement further. This step is supported by an effects description rules base allowing the effects to be characterized. This base is a reference base containing effects characterization properties. It is composed of two kinds of properties: reference properties and custom properties. A reference property is business independent and can be applied to any process. For instance, a property such as "*an activity uses a resource*", is considered to be a reference property, because it can be implemented in any collaborative process. In other words, reference properties are a repository fixed in the rules base. Conversely, a custom property is business dependant, and can be implemented by partners themselves. For example, the crisis management process, "*the management of disaster zone activity requires partner consultation for advice and about the potential actions to implement*" is only applicable in a crisis management context. Thus, partners have to select rules they want to apply in order to highlight effects induced by the implementation of the submitted collaborative process. For the property given above, the effect, produced by a non verification, is characterized as being harmful. Indeed, disaster management is concerned by the definition of a safety zone, the installation of a first-aid station, and informing the population. In this way, all resources are involved in this activity in order to avoid problem when this activity is executed.
At the end of this step, the collaborative process model is checked during the *validation* step, which is partially based on the effects that are detected. In this case, either the process is validated or rejected. If it is approved, the partners validate its calendar and/or perform some adjustments before starting to execute it. If the process is rejected, two cases can be considered. On the one hand, the process can be re-configured in order to break down effects into ones that are considered to be harmful to its execution or ones that result in improved execution. On the other hand, the process is overridden and a new one is developed. In both cases, the new configuration (or new process) is re-submitted to the approach in order to detect other effects. For instance, a new configuration can remove one harmful effect, but this configuration may also create and reveal other effects.

6. Conclusion and prospects

Our paper presents an anticipative effects-driven approach to guide and to assist partners in improving their collaborative processes. It entails the verification and adaptation of a particular collaborative process by anticipating potential effects that could occur during the runtime phase. This adaptation and validation have to lead to improved interaction between each object involved in a collaborative process. Although we apply the approach to a collaborative process for crisis management, it can be deployed in any field that requires a collaborative process to be implemented.
The effective implementation of an approach is based on three concepts that are clearly identified: effects characterization, effects modeling, and model enrichment. Effects characterization allows users to describe an effect, including (1) its nature (*i.e.*, its impact on the collaborative process), and (2) its structure (*i.e.*, its composition *via* natural language). The effect is modeled in a more formal way in order to allow its verification through formal techniques. Moreover, when an effect is identified with knowledge contained in the process model, the modeling language must be enriched. For instance, applying the approach to crisis management has required us to enrich BPMN with concepts specifically related to this domain.
The verification of the potential effects on the collaborative process is performed using formal techniques, such as a conceptual graph. This verification leads to the application of transformations. On the one hand, the collaborative process model is transformed into support and fact models, which include concepts and relationships present in the modeling language as well as a model of the process. On the other hand, the effects models are transformed into conceptual graphs in order to be projected (or not) onto the process model. The result of this task is displayed to the actors who notice the nature of the effects.
As far as outcomes go, several remarks are in order. Conceptual graphs only allow us to verify properties that are time independent. It would be interesting to also consider properties that are related to time. Thus, further work is needed to implement formal verification techniques, such as model checking, that would allow this kind of verification: this research has already been undertaken, and is currently being developed (Mallek *et al.*, 2011).
Another point to pursue is the generalization of the approach to other systems. Currently our work focuses on collaborative processes for crisis management. The objective of this generalization is to enable it to be applied to other domains such as industrial systems. Furthermore, our approach considers only the modeling language such as process model. Ultimately, it must also take other modeling languages into consideration.
Other architectures besides that of complex systems may be interested in the anticipative effects-driven approach. Indeed, some architecture has specific properties that do not characterize complex systems. The Systems of Systems (Maier, 1996) seems to be interesting outcomes to develop the approach.
Finally, our approach does not consider human and psychological effects. It would also be interesting to link our research to the social sciences, in order to analyze these aspects.

7. Bibliography

Interoperability Research for Networked Enterprises Applications and Software (INTEROP Network of Excellence). Annex 1 – Description of Work, 2003.

Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications (ATHENA). Integrated Project Proposal – Description of work, 2003.

IEEE. A compilation of IEEE standard computer glossaries. Standard computer dictionary, 1990.

Chen D, Dassisti M, Elveaeter. Enterprise Interoperability Framework and knowledge corpus – Final report. Interop deliverable Domain Interoperability, DI.3, 2007.

European Community Report (ECR) Unleashing the Potential of the European Knowledge Economy, Value Proposition for Enterprise Interoperability, Final Version (Version 4.0), 2008.

Enterprise Interoperability Research Roadmap (EIRR), Version 4.0, R. Cabral, G. Doumeingts, M. S. Li and K. Popplewell edt., 31 July 2006

Couget P., Bénaben F., Lauras M., Hanachi C., Chapurlat V. (2007). Information Systems interoperability as a way to Partners Integration in a Crisis Context, 20th International Conference on Software & Systems Engineering and their Applications, ICSSEA'07, Paris, France, 2007.

Mayer, F. (1995). Contribution au génie productique : application à l'ingénierie pédagogique en Atelier Inter établissements de Productique Lorrain, Thèse de doctorat en production automatisée, Université Henri Poincaré – Nancy 1 [in French], 1995.

Leger, J.B., Iung, B., Ferro De Beca, A., Pinoteau, J. (1999). An innovative approach for new distributed maintenance system: application to hydro power plants of the REMAFEX project. Computers in Industry, vol 38, pp 131-148, 1999.

Estefan, J. A. (2007). Survey of Model-Based Systems Engineering (MBSE) Methodology. INCOSE MBSE Focus Group Report, 2007.

Daclin, N., Chapurlat, V. (2009). An anticipative effects-driven approach for analyzing interoperability problems in collaborative processes. L.M. Camarinha-Matos et al. (Eds): PRO-VE 2009, IFIP AICT 307, pp. 441-448, 2009.

Mann, D. (2002). Hands on systemic innovation. CREAX Press, 2002.

IEC 60812 (2006). Analysis techniques for system reliability – Procedure for failure model and effects analysis (FMEA). Edition 2.0, 2006.

Lowe D, Ng S. (2004). Effects-based operations: language, meaning and the effects-based approach. In proceedings of Command and Control Research and Technology Symposium-The power of information age and technologies, San Diego, USA, 2004.

Smith, E.A. (2002). Effects Based Operations – Applying Network Centric Warfare in Peace, Crisis and War, Center for Advance Concepts and Technology, DoD Command and Control Research Program, 2002.

Batschelet A.W. (2004). "Effects-based operations: A new Operational Model?", USAWC strategy research project, U.S. Army War College, Carlisle Barraccks, Pennsylvania, U.S.A., 9 april 2002.

Ross, D. (1977). Structured Analysis (SA): a language for communicating ideas. IEEE transactions on Software Engineering, 1977.

ICAM (1981). US Air Force Integrated Computer Aided Manufacturing Architecture. Part 2, volume 4 – Functional Modeling Manual, 1981.

OMG (2009). BPMN: Business Process Modeling Notation. Open Management Group, V1.2,(2009) available online at: *http://www.bpmn.org/* (last visited Tuesday, April 12, 2011)

ISO IS 18629-1 (2004). Industrial automation systems and integration – Process specification language – Part 1: Overview and basic principles.

ATHENA Integrated Project (2005). Report on methodology description and guidelines definition. ATHENA deliverable A1.3.1, 2005.

Mallek, S., Daclin, N., Chapurlat, V. (2010). Towards a conceptualization of interoperability requirements. Interoperability of Enterprises Software and Applications (I-ESA'10), Coventry, England, 2010.

Dindeleux R., Berrah L., Haurat A. (1998) Formal modelling of control processes, European journal of operational research, vol.109 pp. 377 – 389, 1998.

Dubois E., Petit M. (1994), Using a formal declarative language for specifying requirements modelled in CIMOSA in Integrated Manufacturing Systems Engineering, P.Ladet and F.Vernadat Eds., Chapman

& Hall, London, 1994.

Le Moigne J.-L (1977). La théorie du système générale – théorie de la modélisation. Presse Universitaire de France (in French), 1977.

Vernadat, F.B. (1996). Enterprise Integration Modeling and Integration : Principles and Applications. Chapman & Hall, London, 1996.

Chapurlat, V., Roque, M. (2009). Interoperability constraints and requirements formal modelling and checking framework. Advances in Production Management Systems (APMS'09), Bordeaux, France, 2009.

Sowa J.F. (1992). Conceptual Graphes as A Universal Knowledge Representation. Computers Math. Applic., Vol. 23, N° 2-5, p. 75-93, 1992.

ATLAS group (2006). ATL: Atlas Transformation Language. ATL User Manual, version 0.7, 2006.

Lamine E. (2001) Définition d'un modèle de propriété et proposition d'un langage de spécification associé: LUSP. Ph.D. Thesis, Montpellier II University [in French], 2001.

Genest, D. (2010). COGITANT Reference Manual version 5.2.6. 2010, *http://cogitant.sourceforge.net/* (last visited Tuesday, April 12, 2011)

Chein M., Mugnier, M.-L. (1992). Conceptual graphs: fundamental notions, Revue d'intelligence artificielle, vol.6, n°4, pages 365- 406, 1992.

Mallek, S., Daclin, N., Chapurlat, V. (2011). An Approach for Interoperability Requirements Specification and Verification. The International IFIP Working Conference on Enterprise Interoperability, Stockholm, Sweden, 2011.

Maier, M.W. (1996). Architecting principles for Systems-of-Systems. 6^{th} Annual international Symposium of the International Council of System Engineering, Boston, 1996.