AGENCE NATIONALE DE LA RECHERCHE ANR	**SocEDA** ***Cloud based platform for large scale social aware EDA***	SocEDA
	ANR-10-SEGI-013	

Document name:	Adaptation Service (AdS) specification according to the workflow engine used in the ESBs
Document version:	0.1
Task code:	
Deliverable code:	
WP Leader (organisation):	
Deliverable Leader (organisation):	
Authors (organisations):	
Date of first version:	01/06/12

Change control

Changes	Author / Entity	Code of version
Creation of the document	Anne-Marie BARTHE-DELANOË Sébastien TRUPTIL Frédérick BENABEN / ARMINES	0.1 (draft)

Table of Contents

1. Introduction

1.1. Purpose

Depending on evolutions of the situation, the predefined processes or workflows need to be changed. To help these changes, the adaptation service of the SocEDA platform is defined.

With its complex-event processing architecture, French project SocEDA can contribute significantly for the deduction of evolution. Concerning the management of the adaptation, it is facilitated by the increased situational awareness provided by SocEDA platform for event processing. In addition to that, SocEDA would ensure a timely and adequate diffusion of information to relevant actors.

This document aims at describing this Adaptation Service's concepts. For that, this document is divided in two parts: first, an overview of the Adaptation Service connected is provided, then a technical description of the Adaptation Service is given.

1.2. List of Acronyms

Acronym	Definition
A	Army
P	Police
O	Office of infrastructures
MF	Meteo France (weather forecast)
RSN	Radiation Survey Network (measures radioactivity)
EDF	Electricité De France (Operates nuclear plants in France)
RNA	Representative of the National Authority (prefect)
F	Firemen
MEMS	Mobile Emergency Medical Service
WSDL	Web Semantic Description Language
ESB	Enterprise Service Bus
JBI	Java Business Integration
SA	Service Assembly

SU	Service Unit
BC	Binding Component
BPEL	Business Process Execution Language

2. Adaptation service in SocEDA

2.1. Objectives

The Adaptation Service in the SocEDA project is a tool designed to adapt the dynamic of the situation, *i.e.* processes, to the evolution of the context. As explain in D2.4.1, to define the evolution of the processes, two steps are needed:

1. *Detection*: this step consists in detecting an evolution of the situation that could not be solved by the ongoing processes or making the current process not relevant regarding the current situation,
2. *Adaptation*: when an evolution is detected, the adaptation step is executed to modify the ongoing processes in order to make them relevant to the current situation.

In a few words, the Adaptation Service has to allow on one hand to detect if the ongoing processes meet the requirements of the current situation, on the other hand to adapt the ongoing processes if necessary.

2.2. Scientific issues

The major scientific issues concern the detection and the adaptation. Concerning the detection, the Adaptation Service must be able to deduce the matching between the processes and the objectives that the processes are supposed to meet. This question is not trivial, as the measure of this gap is not an easy task and depends on the adopted formalism to represent the processes, the objectives and the current situation and also on determined indicators.

Concerning the adaptation part, the Adaptation Service will have to deduce the processes on the base of the description of the situation. It implies also that the workflows are flexible.

3. Our approach to detect evolutions

The aim of this section is to give an overview of the detection approach that allows detecting evolutions. After explaining the kind of possible evolutions, deducing approach based on events is presented. This deducing approach is based on events managed by the SocEDA Platform. Indeed these events provide information needed to deduce the situation. Therefore in this section, we will present the type of events needed to deduce evolution and then we will explain how detect an evolution.

3.1. Types of evolutions

In (Pingaud, 2009), the author proposes a classification of evolutions, which can occur in the context of collaboration between several heterogeneous organizations. This classification can be adapted as follows:

1. The context itself changes,
2. The set of stakelhoders changes,
3. Dysfunction can append during processes execution.

The following paragraph gives an instance of this classification for a specific context. In crisis management context, this classification could be seen as: (i) the crisis itself evolves (crisis is by nature an instable and evolutionary phenomenon —e.g. an earthquake and its subsequent aftershocks, a flood and the resulting epidemia and famine—), (ii) if the crisis situation does not evolve but the set of partners changes (e.g. Emergency Medical Services arrives, Red Cross leaves), (iii) if a dysfunction happens during the crisis response (fire is not extinguished and fire is going to expand).

3.2. Typology of event: information needed to detect evolution

Information about the situation is needed to detect the new characteristics of the situation that could signify an evolution of the situation.

Concerning activity, several kinds of information are needed: first of all, the relationships between the activities (or the processes) and the characteristics of the situation. Indeed an activity or a process is always realized with the aim to perform an objective.

In a crisis context, activities are realized to prevent a risk or reduce a fact. Therefore, information about the situation could define on one hand the objective and so the execution of activities or processes, on the other hand could check the result of the execution of activities or processes (in other words: if the produced effect is the expected effect). Indeed two kinds of information coud be deduced for each characteristic of the situation: the real state and the deduced state. The real state corresponds to the real situation, and is updated thanks to information coming from the field. The deduced state comes from the state at time zero and inferred by the progress of the activities and processes execution.

Then information about resources are needed. Actually, some activity could only be realized if and only if certain resources are available. Finally information about the result of the execution of activities are needed to detect dysfunction during execution of activities or processes.

To conclude, the deduction of evolution could be realize thanks to the knowledge about:

1. Situation,
2. Activity state,
3. Resource state,
4. Result of activities.

As the SocEDA project is based on an event cloud computing, the SocEDA Platform, all the information needed to detect an evolution are brought by events. Consequently, four kinds of events are defined to use the adaptation service:

1. Situation event,
2. Activity event,
3. Resource event,
4. Consequence event.

The payload and specific name of these kinds of events depends on the use case. Nevertheless, each type of these events has to exist to allow the use of the adaptation service. The following table shows the mapping between the previous event types and the event types used for the nuclear crisis use case.

Table 1: Mapping between adaptation service event types and nuclear crisis use case event types.

Events used by the adaptation service	Events used in the nuclear crisis use case
Situation event	Measure event Alert event
Activity event	Activity Status event
Resource event	Demand event Offer event Resources Status event
Consequence event	Report event Instruction event

3.3. Principles of the approach to detect evolutions

The approach to detect evolutions is based on several steps:

1. Define the current model,
2. Define the reference model,
3. Make a comparison between reference model and current model,
4. Verify the adequacy between the current model and the set of running activities and processes.

All these steps are developed in the sub-sections below.

3.3.1. The current model

As explain in the previous section, it is necessary to follow the information to detect any evolution. Therefore the approach to detect evolutions is based on a model of situation, called *current model*, which has to be updated at each new event reception.

Thus, the current model corresponds at any time to the real situation. The Figure 1 represents the evolution of the current model. At T0, the current model is empty. Then some events are received like the beginning of the service A, an alert about the radiation rate. Accordingly, the current model is updated at each event. At T1 the current model is equal to the situational model at T1 and at T2, the current model diverges from the situational model at T1.

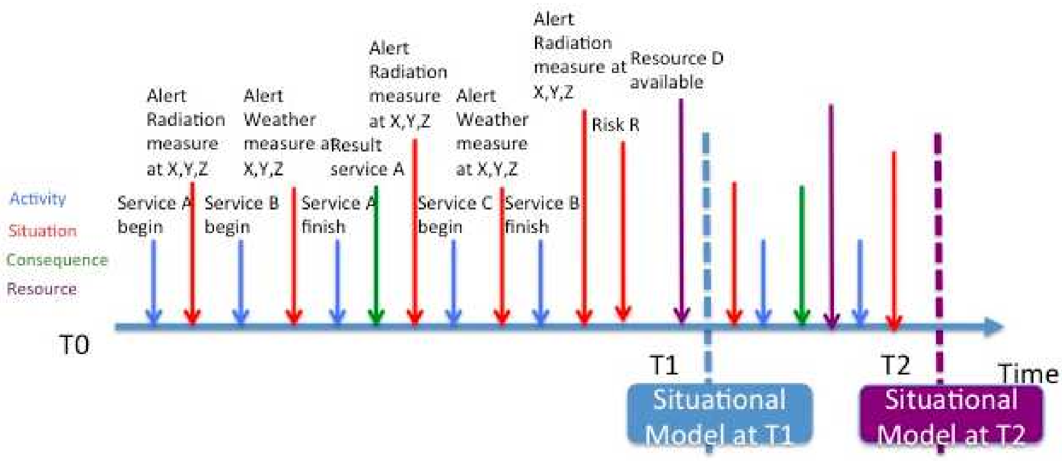

Figure 1: evolution of the current model depending of event.

3.3.2. The reference model

The choice of the set of activities and processes selected to achieve the objectives of the situation is made on the base of a picture of the situation at a specific time. In other words, the choice of

activities and processes realized in T1 could achieve the objectives of the situation at T1 whereas it is impossible for this same set of activities and processes to achieve the objectives of the situation at T2.
Nevertheless, between T1 and T2, some evolutions of the situation are expected. Indeed the execution of activities and processes, selected at T1, has to modify the situation in order to achieve their objectives.
Consequently, it is necessary to make a difference between the expected evolution of the situation and the evolution that implies an adaptation of the set of activities and processes. Thus, a *reference model* is defined. This reference model is a picture of the expected situation in function of the progression of the activities and processes' execution. For instance, in a crisis context, when the activity "extinguish fire" is done, the characteristic "fire" is removed from the reference model.

Thereby, at any time, it is possible to make a comparison between the current model and the reference model to detect an/several evolution(s).

3.3.3. The comparison between current model and reference model

During the execution of activities and processes, both current model and reference models evolve. And these two models evolve by different ways. Indeed the current model evolves with the new information about the situation brought by events, whereas the reference model evolves with the information coming from the progression of the execution of activities and processes.
The Figure 2 illustrates the several evolutions of the models. When the divergence between the two models is considered as too important, an adaptation may be needed.

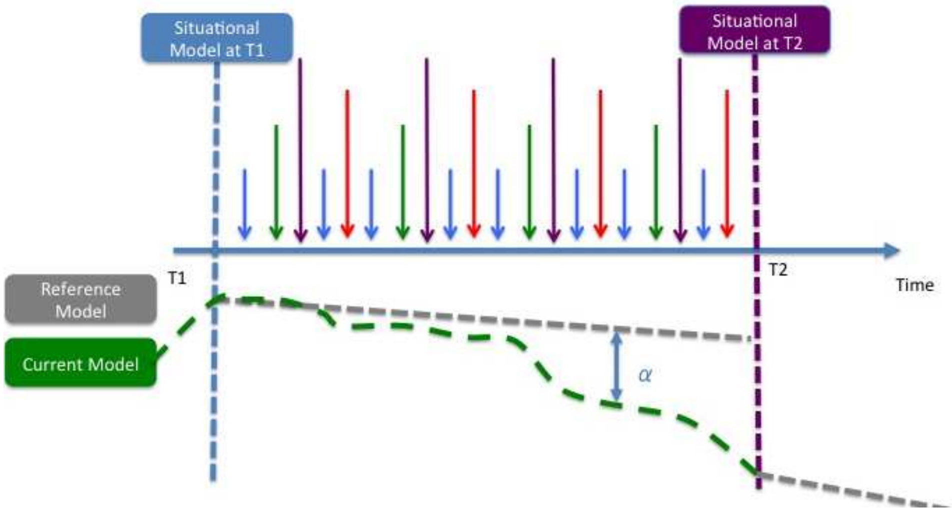

Figure 2: Divergence α between the current model and the reference model.

For the moment, we have created the two models and we ask to the user to decide when the divergence between the models is too important. Therefore, the detection of divergence is manually done. Future works about the calculation of distance between models is needed to automate this step.

When the divergence between these models is estimated too important, a verification of the adequacy between the current model and the set of activities and processes is needed.

3.3.4. Verify the adequacy between the current model and the set of activities and processes.

When the divergence between the current model and the reference model is considered as too important, the last step of the approach for deducing evolutions consists in verifying the adequacy between the current model and the set of activities and processes that are running. Indeed, even if the current model has diverged from the reference model, the set of activities and processes selected could eventually perform the objectives of the new current model.

This step is based on a knowledge-based system. This knowledge-based system is composed of an ontology and a set of adequacy rules.

The ontology gathers information about the relationships between the objectives of the situation and the activities. In a crisis context, the ontology is based on the crisis metamodel defined in (Truptil, 2008).

The adequacy rules verify if the set of activities and processes could perform the situation objectives embedded inside the current model. An adaptation is needed only if the actual activities or processes could not perform at least one objective.

4. Principle of the approach for adapting the activities and processes

The adaptation part of the Adaption Service is divided in two steps:

1. Deduce the needed adaptation,
2. Configure the workflow engine.

4.1. Deduce the needed adaptation

Four kinds of adaptation are possible – the reader can see the D2.4.1 document for more information:

- By design,
- By deviation,
- By underspecification (like Late Binding or Late Modelling)
- by change.

The adaptation service design for SocEDA offers only an adaptation by design. This step is based on results explains in (Truptil, 2011). These results explain how deduce a collaborative process from a characterization of a crisis situation thanks to a knowledge-based system.

The deducing of the needed adaption is thus based on a knowledge-based system. This knowledge-based system is composed of an ontology and deducing rules. The ontology gathers information about the relationship between the objective of the situation and the activities and the deduction rules work like this:

1. for each objectives of the current model, activities or processes that could reach it are selected.
2. For each activities or processes selected during the step 1, if it needs some requires or implies, activities or processes that can reach these needs are selected.
3. Depending on the step 2, the execution order of activities or processes is deduced.

4.2. Configuring the workflow engine

When the set of activities or processes is deduced, the whole of processes are send to the transformation adaptation service, which is a local service deployed in each ESB. This service has to transform the process in BPEL file to allow the workflow engine to execute the new process.

This service is based on an model driven engineering define in (Truptil, 2011).

5. Integration of the Adaptation Service on the SocEDA Platform

The architecture of the SocEDA Platform, as presented in Figure 3 is composed of:

- The **SeaCloud**, containing the dCEP, the Social Filter, the Mashup components and the GUI made by Orange Labs,
- A **pool of ESBs** which embed a workflow engine (orchestrator) running the workflows (the executable version of the defined processes) by calling the partners and ESB's web services (WS). The web services send events to the SeaCloud,
- The **Visualization tool**, plugged on the SeaCloud and allowing human users to have a look on the data coming from the four macro types of events (situational, activity, resources, and consequences).

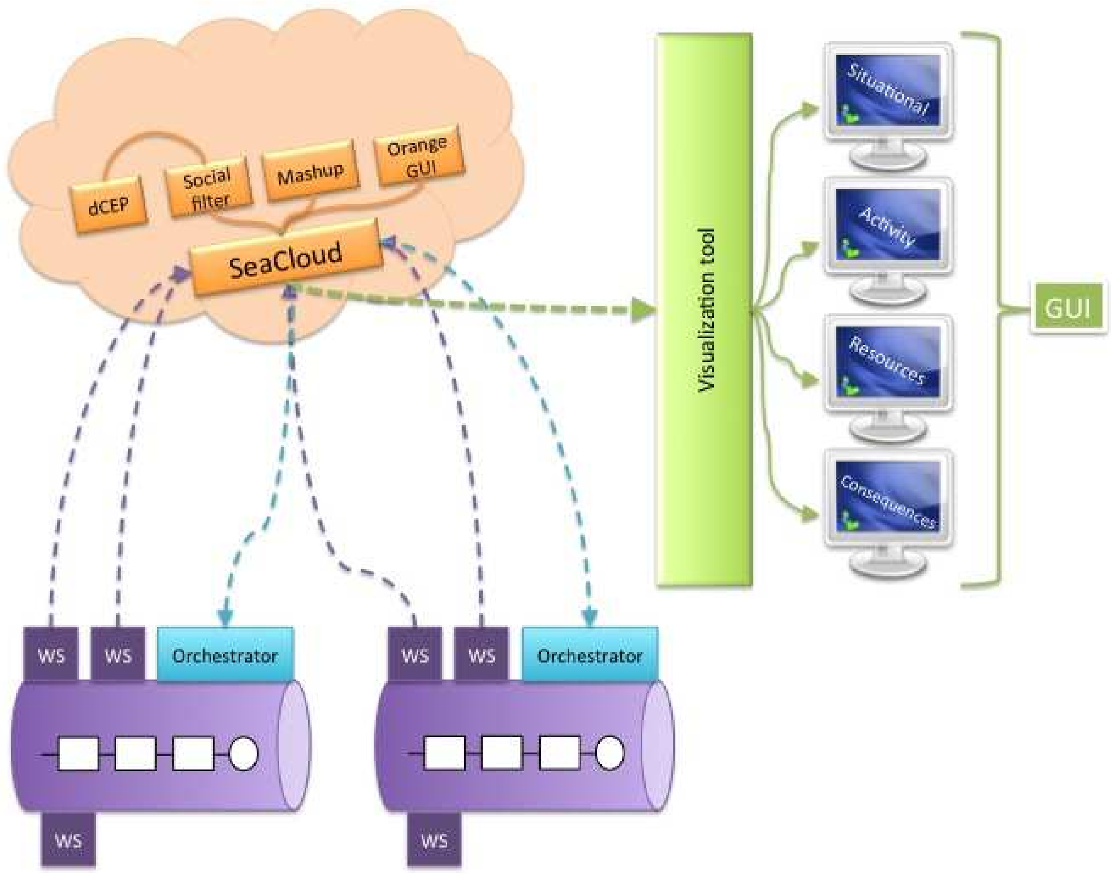

Figure 3. The SocEDA Platform global architecture.

The Adaptation Service is a set of services executing first the Generic Detection, and then the Adaptation of the processes itself.

The Generic Detection step (see Figure 4) consists in first updating the current model with events coming from the SeaCloud (generated by the context of the situation and also the results of the execution of the processes), and the reference model with the data coming only from the monitoring of the execution of the activities and processes.

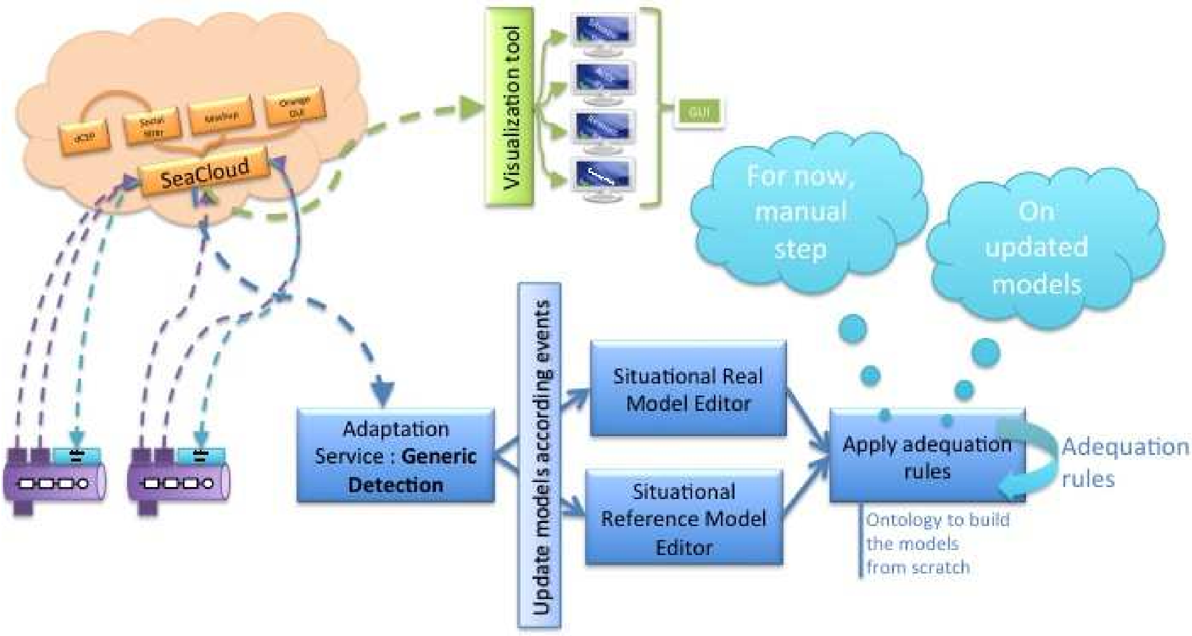

Figure 4. First step in the Adaptation Service execution: the Generic Detection.

Once both models are updated, adequation rules (as seen in section 3.3.3) are applied to detect a divergence between them. For now, this step is manually done but it will be automated in the future.

If no divergence is detected, there is no need to adapt the processes. So the Adaptation Service does not go further.

If any divergence is detected (as illustrated on Figure 5), the Adaptation Service switches on the deduction of the new processes on the base of the current actors'abilities.

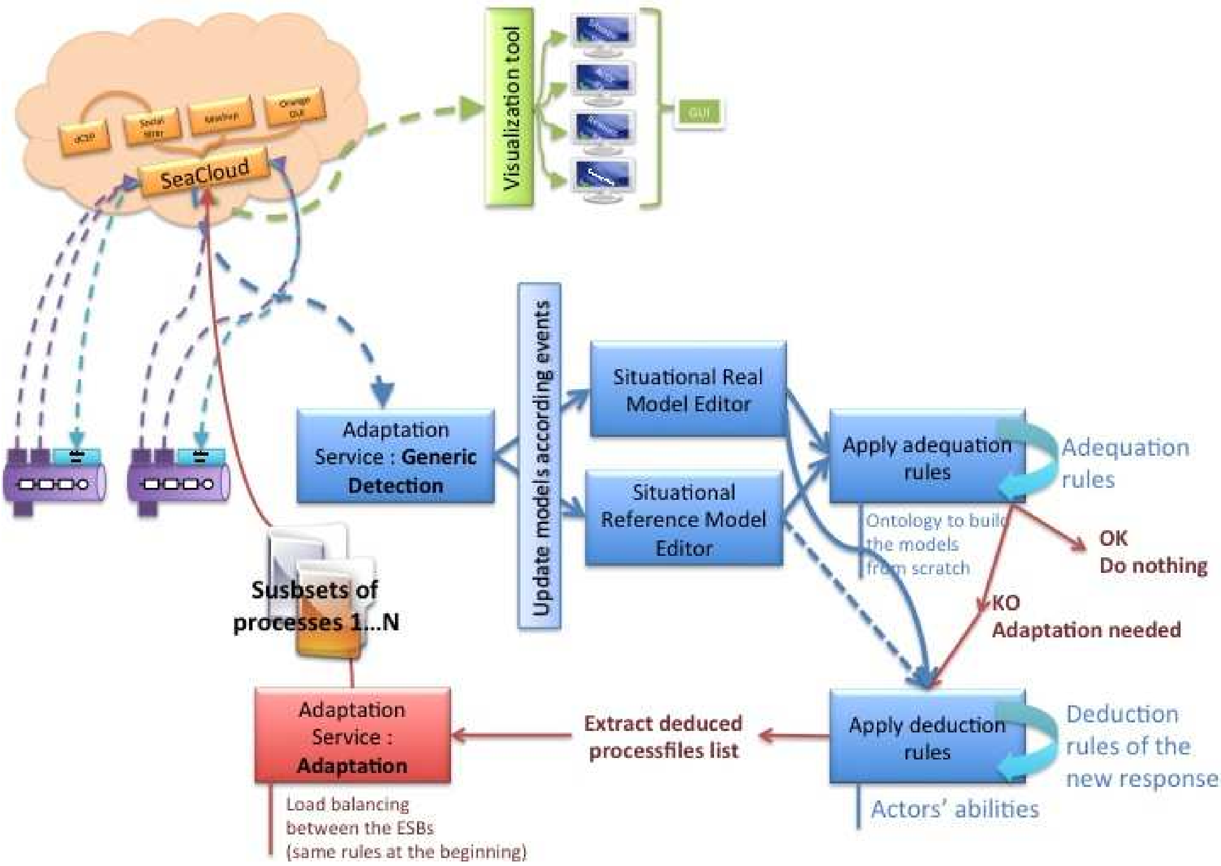

Figure 5. Detection of a divergence and adaptation.

The set of deduced processes is extracted as a list. Then, this is time for the Adaptation step (red box on Figure 5): on the basis of the load balancing rules decided as the initialization of the SoceEDA Platform to run the processes, the new deduced and adapted processes are shared out on the available ESBs. The information (new load balancing decisions and process files) are pushed to the SeaCloud.

The next part of the adaptation concerns the pool of ESBs (Figure 6).

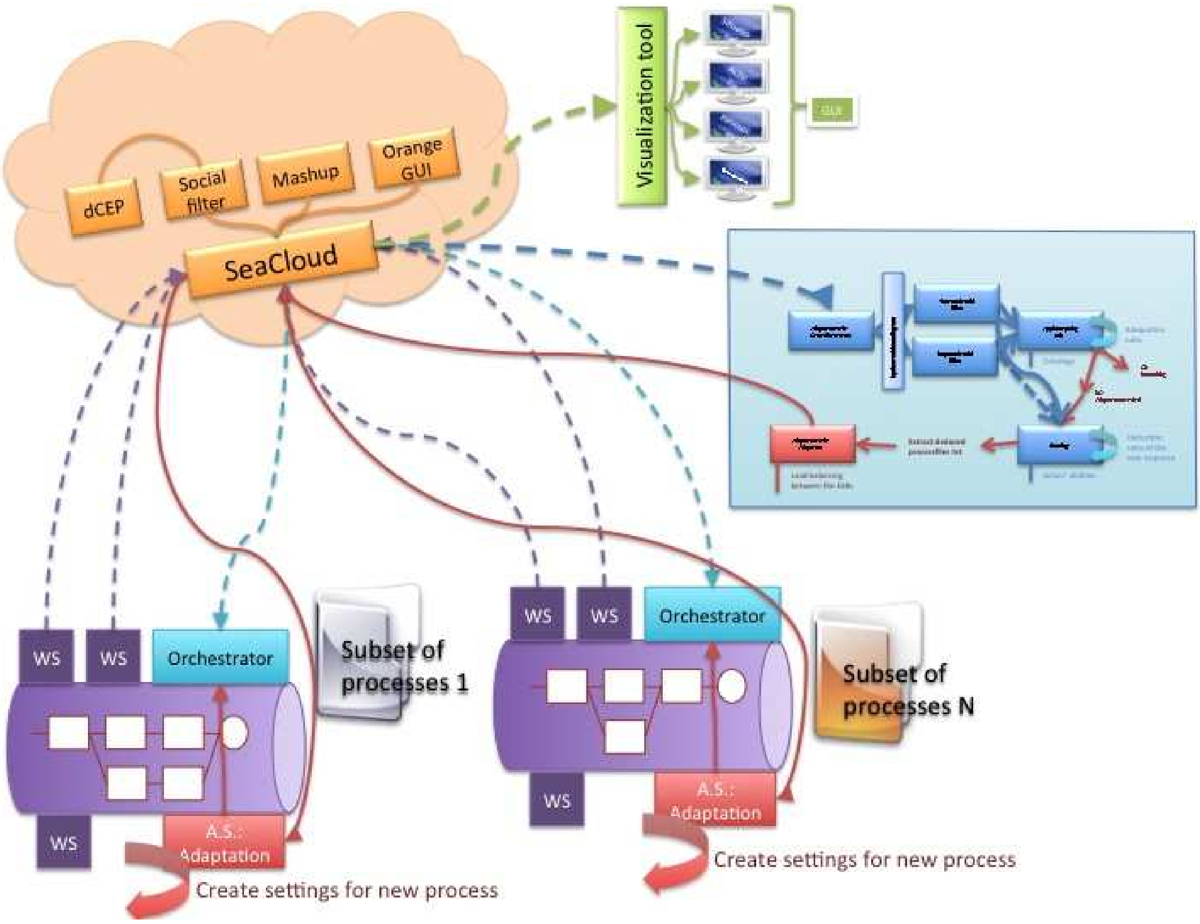

Figure 6. Local Adaptation Service on each ESB.

The SeaCloud transmits the process files to each relevant ESB. These processes are locally turned into executable workflows (*i.e.* written in BPEL because of the orchestrator component, which is a BPEL engine), by the local Adaptation Services (embedded on each ESB). The local Adaptation Service also creates the settings to run the workflow on the ESB where it is hosted. The new workflow files are run by the orchestrator services hosted by the ESBs.

In a few words, each local Adaptation Service manages the implementation of a subset of the processes deduced at the generic level (i.e. in a global point of view where all processes and ESBs are taken in account) on its host.

6. Conclusion

The adaptation service, which is an added value of the Soceda Platform, has to ensure the agility of the dynamic of collaboration in accordance with the evolutions of the context. The objective to ensure agility could be divided in two sub-objectives: (i) how could detect the evolution of the context and (ii) how could adapt the dynamic of the collaboration.

To detect the evolution of the context, two kinds of models are used. The first one, the reference model is a picture of the expected collaboration depending on the progress of the on-going processes. The second one, the current model represents the actual situation. Therefore based on these two kinds of models, it is possible to compare the expected and real situation and when a divergence between these two models is detected an adaptation could be useful.

Once a divergence is, for the moment manually, detected, the adaptation step is launched. The objective of this step is to deduce, from a knowledge-based system, the set of processes needed to achieve the objective of the situation. When this set of processes is deduced, it is splitting up into local service deployed on several ESB in order to ensure a reactive objective of deployment. These local services transform the processes, thanks to a model driven engineering, in a way understandable by the orchestrator engine.

To conclude, this document gives an overview of the adaptation service as well as this connection with the SocEDA platform and the workflow engine.

Future works consist in, first, implemented this adaptation service with the structure explained in this document, second, to define a way to automated the calculation of the divergence between the different models.

7. References

D2.4.1. Detailed SOTA of workflow adaptation/flexibility, delivrable SocEDA project, 2012.

H. Pingaud, *Rationalité du développement de l'interopérabilité dans les organisations*, Management des technologies organisationnelles, Presses de l'Ecole des Mines de Paris, France, pp. 19—30 (2009)

S. Truptil, F. Bénaben, P. Couget, M. Lauras, V. Chapurlat, H. Pingaud "Interoperability of Information Systems in Crisis Management: Crisis Modeling and Metamodeling". IESA'08, Springer, Germany (2008).

S. Truptil, *Etude de l'approche de l'interopérabilité par médiation dans le cadre d'une dynamique de collaboration*, PhD Thesis, Institut National Polytechnique de Toulouse (2011).