A Platform for Event-Driven Agility of Processes: A Delivery Context Use-Case

Anne-Marie Barthe-Delanoë[1], Matthieu Lauras[1], Sébastien Truptil[1], Frédérick Bénaben[1], and Hervé Pingaud[2]

[1] Ecole des Mines d'Albi – Centre de Génie Industriel
Route de Teillet – 81000 Albi, France
{anne-marie.barthe,matthieu.lauras, sebastien.truptil,frederick.benaben}@mines-albi.fr
[2] Université Jean-François Champollion
Place de Verdun – 81000 Albi, France
herve.pingaud@univ-jfc.fr

Abstract. The French SocEDA project aims at providing a Service Oriented Architecture (SOA) platform for dynamic and complex event-driven interactions between large highly distributed and heterogeneous service systems. This platform should be able to combine a publish/subscribe mechanism (to collect events coming from heterogeneous and distributed services) with complex event processing of these collected events in order to detect interesting situations a service should react on. The SocEDA platform should also offer suggestions about relevant changes (adaptation) to do in running processes and services. The context of city logistics is chosen to show some of the basic abilities of the SocEDA platform and the relevance of event management in everyday situations.

Keywords: event-driven architecture, agility, complex event processing, detection, adaptation.

1 Introduction

In the literature of the last decade, many authors have proposed the idea of "city logistics" to solve the issues of urban freight transport like congestion and negative impacts on the environment (such as air pollution and noise) [1][2].

But, as it is widely recognized that companies are nowadays operating in a complex economical environment where markets are more open and globalized, the freight transport companies have also to deal now with uncertainties and short-term changes in demand or supply. They need to have agile processes to respond quickly to these changes, whatever their nature (traffic environment, congestion, safety and energy savings, uncertainties on supply or demand).

This article is organized as follows: Section 2 presents the architecture of the SocEDA platform. Section 3 develops the agility service, including the detection of the relevant moment where change is needed, and the proposed adaptation of running

L.M. Camarinha-Matos and R.J. Scherer (Eds.): PRO-VE 2013, IFIP AICT 408, pp. 681–690, 2013.

processes and services. Finally, Section 4 shows a use case inspired by the delivery of drugs to French pharmacies, with the associated events and business rules.

2 SocEDA Platform

As the SocEDA platform is a global (Internet level scale) structure to combine events from many sources with the goal of connecting and orchestrating services, things and people, it can contributes to support the coordination of shippers, carriers and movements in an integrated logistics system. Through a federated middleware, the platform provides a transparent and distributed infrastructure to allow users and services to be connected in a publish/subscribe mechanism through the whole SocEDA platform. We distinguish the main components that form the federated middleware and the components that are plugged in and using this middleware.

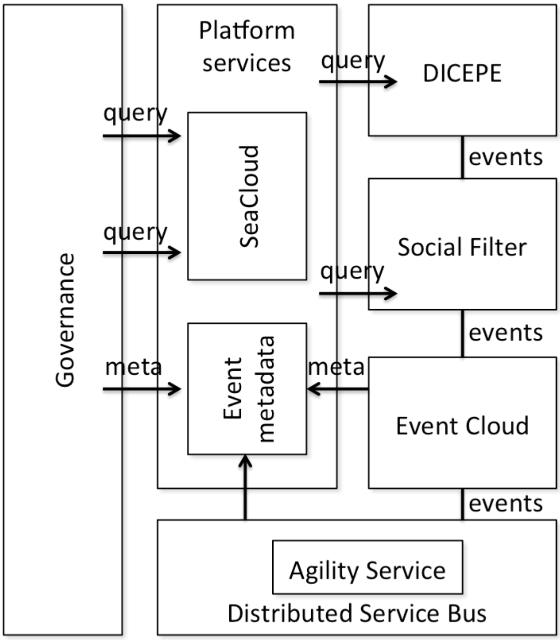

Fig. 1. Overview of SocEDA platform's components

As shown in Figure 1, the main components of the SocEDA platform are:

- Event-Cloud stores and forwards events between the bus and DiCEPE (Distributed Complex Event Processing), which are filtered by the social filter. Every event in the platform must be forwarded to the Event-Cloud to be stored and to be handled asynchronously (pub/sub mechanism) or synchronously by simple query retrieval.
- The Distributed Service Bus (DSB) is in charge of connecting Web Services consumers and providers to the platform (SOA layer). It includes an Event

Broker to integrate events. The DSB implements the Web Service Notification (WS-N) standard [3] in order to provide the core EDA feature.

- The Agility Service has to ensure the agility of the dynamic of the collaboration, in accordance with the evolutions of the context. Its objective can be divided into two sub-objectives: (i) detect the evolution of the context and (ii) adapt the dynamic of the collaboration to the new context.
- The Governance component to govern the services and events.
- SeaCloud allows to subscribe to a specific event producer, to add Complex Event Processing rules or deploy process.

The components of the platform that connects to the federated middleware are:

- The DiCEPE component: reasons on incoming events to generate complex events to the Event-Cloud to be sent to subscribers. It contains a Complex Event Processing (CEP) engine. It detects relevant events or combinations of events by applying event patterns (as described in [4]) on these events.
- Event Sources, i.e. business services in charge of sending/receiving events to/from the federated middleware (partners' devices, sensors, reports, etc.).

3 Agility Mechanism

The delivery processes defined in the city logistics area are subject to issues related to the dynamic adjustments of vehicle, as underlined by [5]. For a decade, several commercial products and research projects have been attempting to provide agility to collaborative workflows. On the commercial side, we can cite Bonita (a suite of tools to design, execute and monitor processes) and the Architecture of Integrated Information Systems (ARIS) tools. ARIS has the ability to combine determined process fragments according to received events. In a way, the ARIS approach manages workflow adaptation (but in a determinist manner) [6]. On the research works side, several projects like the WORKPAD project [7] and the CRISIS project [8] focus more on recovering the disconnecting nodes through specific tasks (WORKPAD project) or supporting collaboration into crisis situation and on exploring decision-making under conditions of uncertainty (CRISIS project). Another research project, the European PLAY project proposes an adaptation recommender service [9] which allows to adapt the ongoing processes on pre-determined milestones, through the addition of relevant pieces of processes (extracted from a knowledge database).

The Agility 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. In this section, we will detail the mechanisms of the Agility Service. The adopted approach of agility in our research work is given by the formula below (which is close to the definition given by [10]):

$$\text{Agility} = (\text{Detection} + \text{Adaptation})*\text{Reactivity}\ . \quad (1)$$

- *Detection:* this step consists in detecting an evolution of the situation that could not be solved by the ongoing processes or making the running processes not relevant to the current situation,
- *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,
- *Reactivity:* detection and adaptation steps have to be done as fast as possible (and real-time if possible).

Agility allows 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.

3.1 Detection

The approach to detect a divergence is based on three major steps:

1. Define the situation model to obtain expected situation model (picture of the expected situation) and field situation model (picture of the ongoing situation),
2. Comparing expected situation model and field situation model,
3. Calculate the difference between these two models to check the adequacy of the running processes with the field situation model.

A situation model is an instant capture of the running collaborative processes, the collaboration itself (all the actors and their services, their objectives, etc.) and the environment in which the processes are running (risks, opportunities, goods, etc.).

We can use these events to track the changes inside the collaborative situation model by this method (illustrated by Figure 2):

- First, we duplicate the initial model of the collaborative situation (i.e. model at time 0),
- Then, both models are automatically updated with the received events (the Agility Service subscribes to all the events emitted by the DiCEPE). We obtain two models through this update:
- The *expected situation model*: the planned and expected situation model at time t (i.e. what we expect to obtain when we apply the collaborative workflows to the collaborative situation). It is obtained by updating the initial model with monitoring events,
- The *field situation model*: the real situation model of the collaboration at time t, whatever the applied collaborative workflows are (i.e. the "what actually happened" situation at time t). It is obtained by updating the initial model with events coming from the field.
- At any time t, a measure of the divergence ∂ is made between the expected model and the field model.

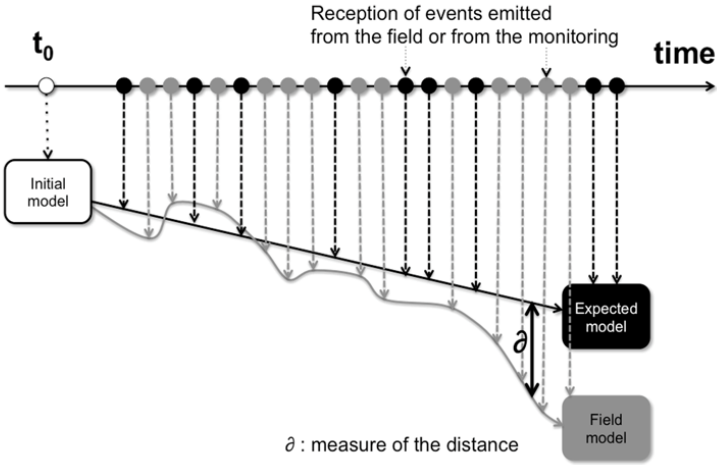

Fig. 2. Principles of detection of the divergence

The measure of ∂ is given by the following formula:

$$\partial = \sum_{i=0}^{n} w_i \partial_i \quad (2)$$

Where:

- ∂_i is a difference identified between the expected model and the field model. E.g.: a partner is not available, a new risk has appeared, a resource is free,
- w_i is the weight applied to ∂_i according the type of element concerned by the identified difference (e.g. partner, risk, resource, activity, etc.) and the kind of difference, called operation here (added, deleted, updated). This weight is used to qualify each detected difference, as each difference has not the same impact on the relevancy of the processes. For example, the addition of an order (the deletion of a partner) has more negative impact on the processes than the deletion of an order (the addition of a new partner) (see Table 1).

The matrix {element type × operation} is a parameter defined by the partners of the collaboration among the execution of the Agility Service. Among the difference detection, a threshold (named $\partial_{\text{threshold}}$) is also defined by the partners. If the calculated ∂ is equal or over this threshold, the Agility Service will execute the adaptation step.

Table 1. Sample of an element types × operation matrix

Element	Added	Deleted	Updated
Partner	1	3	2
Traffic Jam Constraint	4	3	2
Order	2	1	1

3.2 Adaptation

We propose combining two kinds of system adaptations (to the most appropriate conformation): (i) the ability to evolve in a predetermined closed geometry and/or (ii) the ability to redesign a new structure fitting the situation.

The first point refers more to a “Design-time” agility, based, for instance, on risk studies and leading to the building of models including a number of conditional branches to optimize coverage of the possibilities. The second point refers more to a “Run-time” agility where the (re)building of the best possible conformation has to be improvised, at the convenient moment.

To decide what kind of adaptation should be proposed to the user, the adaptation part is based on the study of the difference details gathered by the ∂ calculus in the detection step. First of all, we defined a set of business rules describing how the adaptation step can advice users on the loop to choose (partial or total redefinition, etc.), based on the details of each detected difference and on the weight of each concept participating into the calculated ∂.

Then, when all theses rules are run, the Agility Service is able to indicate the best solution(s) for adaptation to the users, considering all the detected differences. It is very important to note that the final choice is let to the users. The Agility Service is a decision support system in order to choose a relevant adaptation of the processes, not the system that takes the decision of the adaptation.

4 City Logistics and Transportation Use-Case

Our use-case is based on the process of French drug deliveries to pharmacies. It is inspired by the one described in [11]. In France, drugs and Over-the-counter (OTC) medicines are produced by various pharmaceutical manufacturers and mainly distributed through a two-levels network composed of wholesalers and pharmacies (about 80% of the national sales of drugs and OTC medicines [12]) as shown on Figure 3.

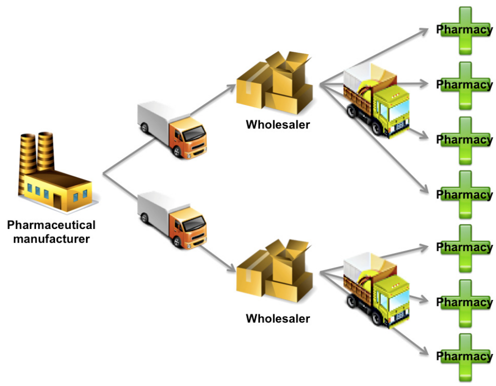

Fig. 3. French drugs and OTC medicines distribution network (according [11])

As the French law requires wholesalers to have a minimum stock of each product in their warehouses, it is considered in this use-case that there is no shortage at wholesaler level. The focus is put on the products distributed from wholesalers to pharmacies. A wholesaler supplies several pharmacies, contained in a given geographical area, about three to four times per day (in average) [12]. The average delay between the reception of the order by the wholesaler, and the delivery to the pharmacy is 2h15. Usually, the deliveries are organized as a set of scheduled rounds.

Considering the fact that the deliveries are dependent on traffic conditions, lorries conditions and availability, weather conditions, it is crucial to (i) gather relevant information in real time and (ii) to manage it very quickly. Thus any changes, any evolution, any information that could challenge the rounds and the quality of service of the delivery process have to be managed in order to be able to adapt the routes of the wholesaler's lorries in real time. According to [4], [13] and [14], these elements that happened and that embedded data can be considered and managed as events.

We have listed four macro types of events covering the whole kinds of events that can happen in various contexts as delivery management, as shown in Table 2. These events help us to gather knowledge about the evolution of the situation (i.e. process execution and environment of execution).

Table 2. Macro event types

Event type	Description
Situational	Used for information about the situation (e.g. information coming from sensors or human made description)
Consequence	Used to transmit the result of one activity/a sequence of activities
Activity	Used to give information about the state of a service: waiting, work in progress, done, failed
Resource	Used for information about the resources (i.e. information such as reports, or objects or human means)

In the context of the delivery use-case, these events can embed at least (i) the traffic information provided by the city hall to assess the traffic situation and potential new routes (situational event), (ii) the pharmacies' stock to assess the priorities in terms of supply (resource event), (iii) the patients' urgent orders to assess the emergencies in term of demand (consequence event), (iv) the ongoing preparations at wholesaler's warehouse to adapt the time delivery schedule (activities event), etc.

Based on these events, we defined some business rules in order to be able to adapt the behavior of the delivery system within the SocEDA platform. Below is a sample of the rules used in this use-case:

- **If** there is an important traffic jam 1 km around $pharmacy_N$ **then** go to $pharmacy_{N+1}$,
- **If** the planned route is modified **then** inform all the remaining pharmacies for the delay,
- **If** there is an urgent request for a specific drug **then** proceed to the delivery of this drug first.

The considered scenario concerns one vehicle, which has to deliver five pharmacies (P) located in various places of a given area. The scheduled round is:

1. Start from wholesaler's warehouse at 8:30 am,
2. Arrive to P#53 at 8:45 am and leave at 8:55 am,
3. Arrive to P#22 at 9:15 am and leave at 9:25 am,
4. Arrive to P#34 at 9:30 am and leave at 9:40 am,
5. Arrive to P#11 at 10:00 am and leave at 10:10 am,
6. Arrive to P#14 at 10:20 am and leave at 10:30 am,
7. Go back to warehouse at 10:50 am.

The wholesaler has chosen a $\partial_{\text{threshold}}$ equals to 2.

At 8:25 am the vehicle is ready to start. Its round begins at 8:30 am. At 8:43 am, the driver arrives to P#53 and leaves at 8:51 am.

While he is driving to P#22, he received an alert about a traffic jam due to a street protest in the street just near P#22. **The Agility Service of the SocEDA platform adds a *Traffic Jam Constraint* element into the field situation model (and nothing into the expected situation model). According the matrix {element type × operation}, $\partial = 4$ and $\partial > \partial_{\text{threshold}}$.** So the Agility Service of the SocEDA platform modifies its round schedule and proposes an adapted route to the driver, in order to avoid the traffic jam and to deliver P#34 instead of P#22. The SocEDA platform also informs P#22 that the delivery should be effective around 10:45 am. Finally, he arrives at P#34 at 9:18 am and leaves at 9:28 am.

The new schedule indicates to the driver the route to go to P#11. While he is driving to P#11, he receives an alert about an urgent demand received by P#14 and for which the medicines are in the delivery order he is transporting. **The Agility Service of the SocEDA platform adds an *Order* element into the field situation model (and nothing into the expected situation model). According the matrix {element type × operation}, $\partial = 2$ and $\partial = \partial_{\text{threshold}}$.** Consequently, the platform suggests him to change his route to deliver first P#14 instead P#11. And the pharmacist of P#11 receives a notification from the platform at 9:38 am about the move of its delivery (from 11:00 am to 10:25 am). Finally the driver reaches P#14 at 9:58 am and leaves at 10:13 am. The medicine is delivered without any delay to the patient. The driver continues his route normally and arrives à P#11 at 10:24 am and leaves at 10:35 am.

The driver goes to P#22 following the route proposed by the platform: he reaches P#22 at 10:45 am and leaves at 10:55 am.

Finally the driver goes back to the wholesaler's warehouse at 11:05 am, with only a delay of 15 min on the planned schedule. This example shows how the SocEDA platform allows detecting, analyzing and reacting on events and can help people avoiding losing time and means by adapting processes.

5 Conclusion

The SocEDA platform provides an environment to manage processes in an event-driven perspective, allowing the users to be dynamically connected to events coming from the environment and the processes themselves. Events are gathered through a publish-subscribe mechanism based on topic and content subscription. This kind of subscriptions is very interesting because it allows the user to get the relevant information at the right time, without any necessity of knowing all the event sources providing this information. Events gathered by the SocEDA platform are filtered and aggregated to detect interesting situations and, through the SocEDA's Agility Service, to propose relevant changes to existing and running processes and services.

Regarding the overall proposition, two reproaches might be done: (i) what about the robustness of such a system? And (ii) how to deal with access to events and unreliable data?

Concerning the first point, it is clear that as we are based on organizations' Information Systems, we are thus technically dependent on the network. If the network goes down, a part or, in the worst case, the entire system fails too. Hardware security measures should be taken to protect the physical network, in addition to the security measures taken to protect the data network against acts of piracy.

Concerning the second point, we agree on that events can only be used if they are trustable: a possible solution to tackle this point may be the definition of a social filter to evaluate the event providers and the events themselves [15] [16]. This social filter operates on a social network of services to compute the strength of the relationships between them through a trust interference algorithm. The social filter is one of the components of the SocEDA platform and will be integrated on the CEP. Another complementary solution could be the use of a governance tool to manage subscriptions to event types.

Acknowledgments. This work has been partially funded by the French Research Agency (ANR) regarding the research project SocEDA (SOCial Event Driven Architecture) (Grant ANR-10-SEGI-013). This project aims to provide dynamic and adaptive workflows to collaborative situations through EDA and CEP. The authors would like to thank the project partners for their advices and comments regarding this work.

References

1. Ruske, W.: City-Logistics - Solutions for urban commercial transport by cooperative operations management. Presented at the OECD Seminar on Advanced Road Transport Technologies. Proceedings. TT3 (June 1994)
2. Taniguchi, E., Thompson, R.G., Yamada, T., van Duin, R.: City Logistics: Network Modelling and Intelligent Transport Systems. Pergamon (2001)
3. OASIS: Web Services Base Notification 1.3 OASIS Standard (2006), http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf
4. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications Co., Greenwich (2010)
5. Taniguchi, E., Shimamoto, H.: Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. Transportation Research Part C: Emerging Technologies 12, 235–250 (2004)

6. Scheer, A.-W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management. In: van der Aalst, W., et al. (eds.) Business Process Management. LNCS, vol. 1806, pp. 376–389. Springer, Heidelberg (2000)
7. Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Russo, A., Steinmann, R., Bortenschlager, M.: WORKPAD: process management and geo-collaboration help disaster response. IJISCRAM 1, 32–49 (2011)
8. Crisis Project website, `http://idc.mdx.ac.uk/projects/crisis/`
9. Verginadis, Y., Patiniotakis, I., Papageorgiou, N., Stuehmer, R.: Service Adaptation Recommender in the Event Marketplace: Conceptual View. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 194–201. Springer, Heidelberg (2012)
10. Charles, A., Lauras, M., Wassenhove, L.V.: A model to define and assess the agility of supply chains: building on humanitarian experience. International Journal of Physical Distribution & Logistics Management 40, 722–741 (2010)
11. Lauras, M., Verginadis, Y., Stuhmer, R., Benaben, F.: An event-driven platform to manage agility: Behavior adaptation in delivery context. In: 2012 6th IEEE International Conference on Digital Ecosystems Technologies (DEST), pp. 1–6 (June 2012)
12. Le Hénaff, S.: Médicaments, une distribution fébrile (2012)
13. Chandy, K.M., Schulte, W.R.: Event Processing: Designing It Systems for Agile Companies. McGraw-Hill Prof Med/Tech (2009)
14. Luckham, D., Schulte, W.R. (eds.): Event Processing Glossary - Version 1.1 (2008), `http://complexevents.com/wp-content/uploads/2008/08/epts-glossary-v11.pdf`
15. Hasan, O., Brunie, L., Bertino, E., Shang, N.: A Decentralized Privacy Preserving Reputation Protocol for the Malicious Adversarial Model. In: LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon (2012)
16. Hasan, O., Miao, J., Mokhtar, S.B., Brunie, L.: A Privacy Preserving Prediction-based Routing Protocol for Mobile Delay Tolerant Networks. Proceedings of MASS 2012 (2012)