Process Adaptation Patterns for Cross-Organizational Business Process Modeling

Moufida Aouachria[1,2,3], Abderrahmane Leshob[1] and Abdessamed Réda Ghomari[4]
[1]Laboratory for Research on Technology for Ecommerce (LATECE), University of Quebec at Montreal, Montreal, Canada
[2]Division Télecom, Centre de Développement des Technologies Avancées, Algiers, Algeria
[3]Université de Batna 02, Batna, Algeria
[4]LMCS Laboratory, École Nationale Supérieure d'Informatique, Algiers, Algeria

Keywords: Business Process Modeling, Business Patterns, Business Process Integration, Process Adaptation Patterns, Process Mining, Workflow Net.

Abstract: Nowadays organizations collaborate through cross-organizational business processes. These business processes require the coordination of several partners who are often geographically dispersed. Modeling such processes is complex and requires that designers have extensive experience in particular when organizations' processes are incompatible. This paper addresses the problem of modeling cross-organization processes out of collection of organizations private process models. To this end, we propose a set of process adaptation patterns that connect private processes and resolve interoperability issues. Proposed patterns are formalized with workflow net.

1 INTRODUCTION

Nowadays organizations collaborate through business processes that cross organizations' boundaries. These business processes often require the coordination of several organizations that are often geographically dispersed. Cross-organization processes must take into account collaborative scenarios involving distributed and autonomous partners. For example, a typical 'Sales & Distribution' process requires the coordination of several business partners including the buyer, the supplier, the carrier, and other related partners such as financial institutions. Modeling such processes is complex and requires that designers have extensive experience (Zeng et al., 2013).

Coordination complexities come from the fact that 1) organizations use their business processes to collaborate with multiple partners who may have different business models (i.e., way of doing things), 2) private processes that need to be connected may be incompatible (e.g., structural mismatch between messages), and finally 3) these processes are often supported by have information systems. Coordination complexities have been addressed much more at software level (e.g., web services) through message-based patterns (e.g., (Wang et al., 2007; Li et al., 2010)). Such patterns are the focus of our processes adaptation operators.

This paper proposes a set of workflow patterns we call *process adaptation patterns* that aim to: 1) help modeling cross-organizational processes, 2) resolve interoperability issues between partners' processes, and 3) improve process flexibility. Proposed patterns are formalized with *workflow net* (Van der Aalst, 1998), a formal language for modeling workflow processes.

The rest of the paper is organized as follows: Section 2 introduces the problem and the motivations of this work. Section 3 provides an overview of our approach and describes our research vision. Section 4 presents the process adaptation patterns. Section 5 surveys related work. We conclude in section 6.

2 THE RESEARCH PROBLEM

Consider two companies: *Company A* (Buyer) wants to collaborate with *Company B* (Supplier). *Company A* buys its products using the (private) *procurement* process of Figure 1. *Company B* sells its products using the (private) *Sales and Distribution* (SD) process of Figure 2.

As shown in the BPMN model of Figure 1, the

Aouachria, M., Leshob, A. and Ghomari, A.
Process Adaptation Patterns for Cross-Organizational Business Process Modeling.
DOI: 10.5220/0006470201240129
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 2: ICE-B*, pages 124-129
ISBN: 978-989-758-257-8

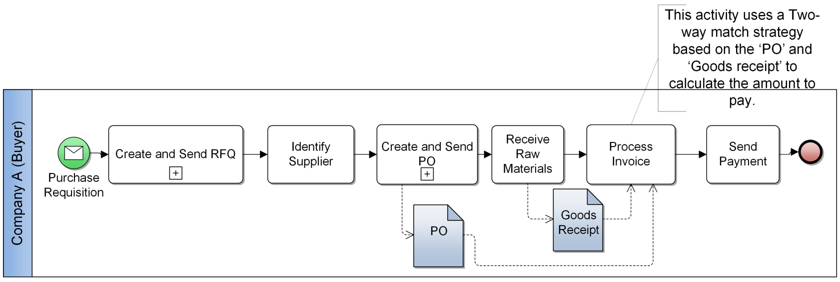

Figure 1: Procurement private process.

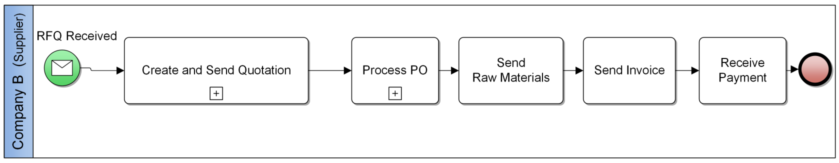

Figure 2: Sales and Distribution private process.

procurement process starts when it receives a purchase requisition. The process then sends a request for quotation (RFQ) to potential suppliers who, in turn, prepare quotations and submit them back to the company. After receiving quotations, *Company A* selects a supplier, creates a purchase order (PO) and sends it back to that supplier. Once the products are received, a goods receipt is generated and the payment is made.

One of the peculiarities of this procurement process is that the requester (i.e., Company A) does not need to receive an invoice to generate a payment. Indeed, it uses the *Two-way match* strategy for invoice processing. Unlike, the traditional *three-way match* approach consisting of matching invoice, 'PO', and receiving reports, *Company A* uses only the 'PO' and the goods receipt to calculate the invoice value.

On the other hand, the 'SD' process (Figure 2) of *Company B* starts by receiving an RFQ from a purchaser. *Company B* then prepares a quotation and sends it back to the purchaser. After receiving the PO, *Company B* fulfills the order and delivers it to its clients. Once the products are delivered, an invoice is generated. The process ends once the payment is received.

To enable these two organizations to collaborate, we need to link their two private processes to build a single new collaboration process which crosses the organizations boundaries. This is what we call *Cross-Organizational Business Process*. Unfortunately, this process has an interoperability issue. Indeed, the supplier (Company B) sends an invoice which is not expected by the requester (Company A).

This problem has two major issues. First, it makes the new cross-organizational process invalid in the context of this collaboration. Secondly, it will prevent both processes from collaborating using information systems that support them. Such incompatibility problems are the focus of our process adaptation patterns presented in section 4. To resolve the interoperability issue presented in this example, we propose a process adaptation pattern called *Single-Entry-Zero-Exit* (SEZE) (see subsection 4.1) that hides the invoice message sent by the supplier.

Before going into the details of each process adaptation pattern, we will present in the next section an overview of our research vision.

3 OUR RESEARCH VISION

We would like to provide organizations with tools to help them model cross-organization processes that accurately reflect their way of doing things from a collection of private process models. To this end, we propose a three-step approach as illustrated in Figure 3. The first step analyzes the input processes (i.e., private processes of the organizations that plan to collaborate) to identify interoperability issues using a process mining approach. More precisely, we plan to analyze the logs captured by the different systems that support each private process. The second step uses process adaptation patterns specification, we propose in this paper, as input to identify the patterns that resolve interoperability issues identified in the first step. The last step applies identified patterns to adapt input processes (organizations' private processes) and build a single cross-organizational process.

The work presented in this paper deals exclusively with the formalization of process adaptation patterns

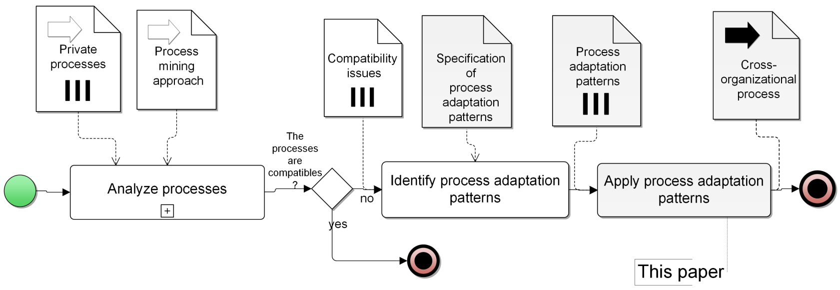

Figure 3: Overall process of the research objectives.

(see Figure 3) we derived from the service oriented architecture (SOA) literature (see (Wang et al., 2007; Li et al., 2010)).

4 CROSS-ORGANIZATIONAL PROCESS ADAPTATION PATTERNS

The purpose of this work is to assist organizations in the process of modeling cross-organizational processes. This, by resolving incompatibility issues when combining different organization's private processes. To this end, we have identified a set of six process adaptation patterns that resolve message mismatches in process collaboration, namely Zero-Entry-Single-Exit$\left(\text{ZESE}_{adapt}\right)$, Single-Entry-Zero-Exit$\left(\text{SEZE}_{adapt}\right)$, Single-Entry-Multiple-Exits$\left(\text{SEME}_{adapt}\right)$, Multiple-Entries-Single-Exit $\left(\text{MESE}_{adapt}\right)$, Multiple-Entry-Multiple-Exit$\left(\text{MEME}_{adapt}\right)$, and Single-Entry-Single-Exit$\left(\text{SESE}_{adapt}\right)$. Each pattern is a *process fragment* consisting of adaptation activities. For the sake of simplicity, we choose to demonstrate our ongoing work by formalizing two patterns we used the most within our experimental data, namely $\left(\text{SEZE}_{adapt}\right)$ and SEME_{adapt}.

Next, we propose two definitions: 1) a definition of each activity in a cross-organizational process, and 2) a definition of $\text{Adapt}_{\text{WFnet}}$, an extended Workflow net (Van der Aalst, 1998), we propose to model proposed process adaptation patterns. We choose Workflow net language mainly for its formal semantic, expressiveness, graphical nature, and its ability to analyze processes (Van der Aalst, 1998; Zeng et al., 2013). It will be assumed throughout that the reader is familiar with WF_{net}(see (Van der Aalst, 1998; Zeng et al., 2013)).

Definition 1: An activity A in cross-organizational process is defined as an 9-tuple$(ID, ST, ET, Org_{Pro}, Org_{Rec}, M_s, M_r, R_{Rel}, R_{Req})$, where:

- ID, its identification. ST, its start time. ET, its end time.
- Org_{Pro}, a private organization's process that sends message to A.
- Org_{Rec}, a private organization's process that receives messages from A.
- $M_s = \{M_i\ ,\ 1 \leq i \leq n\}$, set of messages sent by A.
- $M_r = \{M_i\ ,\ 1 \leq i \leq m\}$, set of messages received by A.
- $R_{Rel} = \{R_i\ ,\ 1 \leq i \leq n\}$, set of released resources after the execution of A.
- $R_{Req} = \{R_i\ ,\ 1 \leq i \leq m\}$, set of required resources by A to execute.

To formalize our process patterns, we adapted RM_{WFnet} (Zeng et al., 2013) using the following definition:

Definition 2: We call Adapt_{WFnet} a Petri net composed of 4-tuples $(P,\ A;\ F,\ M_0)$, where P is a finite set of places, A a finite set of transitions, F a set of arcs , $P \cap A = \emptyset$, and M_0 is the initial marking of Adapt_{WFnet} where:

1. $P = P_L \cup P_R \cup P_M$, $P_L \cap P_R = \emptyset$, $P_L \cap P_M = \emptyset$, *and* $P_R \cap P_M = \emptyset$; $P_R \subseteq P$ represents the resources in the workflow; and $P_M \subseteq P$ represents the exchanged messages in the workflow; and P_L represents other places which are different from $P_R \cup P_M$.
2. $F = F_L \cup F_R \cup F_M$, where:
   - (a) $F_L = (P_L \times A) \cup (A \times P_L)$, represents the logical structure of the cross-organizational model;
   - (b) $F_R = (P_R \times A) \cup (A \times P_R), \forall x,\ y \in A \cup P_R$, $(x,\ y) \in F_R$ iff $(y, x) \in F_R$. F_R, represents the resource relations of the cross-organizational model;
   - (c) $F_M = (P_M \times A) \cup (A \times P_M)$, $\forall x,\ y \in A \cup P_M$, $(x,\ y) \in F_M$ iff $(y,x) \notin F_M$. F_M , represents the message relations of the cross-organizational model;

3. $A = A_s \cup A_a \cup A_r \cup A_l$, where:
   (a) $A_s \subseteq A$, represents sending activities;
   (b) $A_a \subseteq A$, represents adaptation activities;
   (c) $A_r \subseteq A$, represents receiving activities;
   (d) $A_l = A \setminus \{A_s \cup A_a \cup A_r\}$, represents remaining activities; where:
   $\forall A_i, A_j \in \{A_s, A_a, A_r, A_l\}$ /if $A_i \neq A_j$ then $A_i \cap A_j = \emptyset$.
4. $(P_L, A; F_L)$ is a WF-net.
5. $\forall p \in P,\ M_0(P) = \begin{cases} 1 & if\ p = P_R \cup \{O\} \\ 0 & otherwise \end{cases}$, Where:
   $O \in P_L \wedge {}^\bullet O = \emptyset$. Where:
   ${}^\bullet O = \{y / y \in P \cup A \wedge (y, O) \in F\}$ is the pre-set of O.

As shown in the definition 2 above, the transition set A in an $\text{Adapt}_{\text{WFnet}}$ is used to represent workflow activities using four components, namely A_s, A_a, A_r, and A_l. Compared to $RM_{WF\,net}$ (see (Zeng et al., 2013)), transition set A represents the activities in the whole cross-organizational process and not only in the private processes.

The rest of this section describes two process adaptation patterns using $\text{Adapt}_{\text{WFnet}}$. For each pattern we provide its name, description, an example that illustrates the pattern, the problem, and the solution. To ensure that the resulting formalization is safe, we adapted the mechanism proposed by Wang (Wang et al., 2007). This mechanism allows only one working cycle of the adapter to execute at a time. In our adapter patterns, this blocking mechanism is applied via the usage of resource tokens. Resource tokens are consumed once a working cycle starts, and released when the cycle completes. Resource place is represented by a double-line circle with a token however message place is represented by a double-line circle without a token (Zeng et al., 2013).

4.1 Single-Entry-Zero-Exit Adaptation Pattern $(\text{SEZE}_{\text{adapt}})$

4.1.1 Description

This adapter has the ability to delete a message sent by one process to a target process where the latter does not expect it.

4.1.2 Example

After receiving a purchase order, the provider process sends a receipt while the receiving process does not expect such message. The adapter should just hide or delete this message after the communication ends.

4.1.3 Problem

The source process has an extra message that the target process does not expect to receive.

4.1.4 Solution

When one process sends a message that is not expected by the target process, the adapter should just kept this message, it can be deleted after the communication ends.

Let say that a sending activity A_{si} sends a message M_{si} from Org_{Pro} but this message is not expected by the receiving process (Org_{Rec}). In this case, process adaptation pattern $\text{SEZE}_{\text{adapt}}$ should be inserted to hide this message. Therefore, $\text{SEZE}_{\text{adapt}}$ pattern is defined by two activities $\{A_{si},\ A_{ak}\}$ as follow:

$$\begin{cases} (A_{si},\ ST_{si},\ ET_{si},\ \emptyset,\ Org_{Rec},\ \{M_{si}\},\ \emptyset,\ \emptyset,\ \{R_{reqi}\}) \\ (A_{ak},\ ST_{ak},\ ET_{ak},\ Org_{Pro},\ \emptyset,\ \emptyset,\ \{\mathrm{M_{rk}}\},\ \{R_{relk}\},\ \emptyset) \end{cases}$$

$$\text{Where:} \begin{cases} A_{si}.R_{reqi} = A_{ak}.R_{relk} = R_i \\ A_{si}.M_{si} = A_{ak}.M_{rk} = M_i \\ A_{si}.ET_{si} \prec A_{ak}.ST_{ak}. \end{cases}$$

The $\text{Adapt}_{\text{WFnet}}$ of $\text{SEZE}_{\text{adapt}}$ pattern (See Figure 4) can be defined as a tuple $(\text{P},\ \text{A};\ \text{F},\ M_0)$ where:

$$\begin{cases} P = \{M_i\} \cup \{R_{\mathrm{i}}\}, \\ A = \{A_{si}\} \cup \{A_{ak}\}; \\ F = \{(A_{si}, M_i), (R_i, A_{si})\} \cup \{(M_i, A_{ak}), (A_{ak}, R_i)\}, \\ M_0 = [0, 1] \end{cases}$$

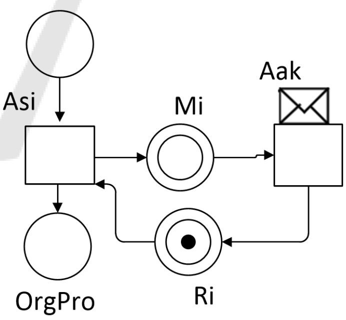

Figure 4: SEZE adaptation pattern ($\text{SEZE}_{\text{adapt}}$) .

This adapter has only a single entry sending activity A_{si} got by place M_i. The message getting from M_i will not exit to receiving activity. Single to zero adaptation is performed by A_{ak}. $SEZE_{adapt}$ pattern starts a working cycle upon the run of A_{si} and completes the cycle upon the run of A_{ak}. Accordingly, the run of A_{si} consumes the resource token in place R_i and the occurrence of A_{ak} returns the token back to R_{i}.

4.2 Single-Entry-Multiple-Exits Adaptation Pattern $(\text{SEME}_{\text{adapt}})$

4.2.1 Description

This adapter has the ability to split a message M_{si} sent by one process while this message is expected by the target process as N fragments of messages $M_{r1}, M_{r2}, \ldots, M_{rN}$.

4.2.2 Example

When a physician writes prescription, then he sends it within a single message, with multiple instructions $(M_{r1}, M_{r2}, \ldots, M_{rN})$ to a blood analysis laboratory. The latter expects to receive it in multiple different messages. The adapter must split the message into multiple messages $(M_{r1}, M_{r2}, \ldots, M_{rN})$ and ensures that all messages are supplied to the blood laboratory.

4.2.3 Problem

A process sends a single message to a target process while the latter expects to receive this message in fragment.

4.2.4 Solution

When a process (Org_{Pro}) sends a message, with multiple information to target process (Org_{Rec}) that expects it in multiple different messages; the adapter splits the message sent by process Org_{Pro} and ensures that all messages are supplied to Org_{Rec}.

Let say that an activity A_{si} sends a message M_{si} from Org_{Pro} but the receiving process Org_{Rec} expects to split it to receive. In this case, process adaptation pattern $\text{SEME}_{\text{adapt}}$ should be inserted to split the message M_{si} into N message $\{M_{r1}, M_{r2}, \ldots, M_{rN}\}$ expected by the receiving process (Org_{Rec}). Then, $SEME_{adapt}$ pattern is defined by activities $\{A_{si}, A_{ak}, A_{r1}, A_{r2}, \ldots, A_{rN}\}$ as follow:

$$\begin{cases} (A_{\text{si}}, ST_{\text{si}}, ET_{si}, \emptyset, Org_{Rec}, \{M_{\text{si}}\}, \emptyset, \emptyset, \\ \{R_{req1}, R_{req2}, \ldots, R_{reqN}\}) \\ (A_{ak}, ST_{ak}, ET_{ak}, Org_{Pro}, Org_{Rec}, \{\text{M}_{\text{sk}}\}, \{\text{M}_{\text{rk}}\}, \emptyset, \emptyset), \\ (A_{r1}, ST_{\text{r1}}, ET_{r1}, Org_{Pro}, \emptyset, \emptyset, \{M_{\text{r1}}\}, \{R_{\text{rel1}}\}, \emptyset), \\ (A_{r2}, ST_{\text{r2}}, ET_{r2}, Org_{Pro}, \emptyset, \emptyset, \{M_{\text{r2}}\}, \{R_{\text{rel2}}\}, \emptyset), \\ \ldots\ldots \\ (A_{rN}, ST_{\text{rN}}, ET_{rN}, Org_{Pro}, \emptyset, \emptyset, \{M_{\text{rN}}\}, \{R_{\text{relN}}\}, \emptyset). \end{cases}$$

where:

$$\begin{cases} A_{rj}.\text{R}_{\text{relj}} = A_{\text{si}}.R_{reqj} = R_j, \forall j,\ 1 \le j \le N, \\ A_{ak}.\text{M}_{\text{sk}} = \bigcup_{\forall j,\ 1 \le j \le N} A_{rj}.M_{\text{rj}}, \\ A_{ak}.\text{M}_{\text{rk}} = A_{si}.\text{M}_{\text{si}} = \text{M}_{\text{si}} and A_{si}.ET_{si} \prec A_{ak}.ST_{ak}, \\ A_{ak}.ET_{ak} \prec A_{rj}.ST_{rj}, \forall j,\ 1 \le j \le N. \end{cases}$$

The $\text{Adapt}_{\text{WFnet}}$ of $SEME_{adapt}$ pattern (See Figure 5) can be defined as a tuple $(P, A; F, M_0)$ where:

$$\begin{cases} P = \{M_{si}, M_{r1}, M_{r2}, \ldots, M_{rN}\} \cup \{R_1, R_2, \ldots, R_N\}, \\ A = \{A_{si}\} \cup \{A_{ak}\} \cup \{A_{r1}, A_{r2}, \ldots, A_{rN}\}; \\ F = \{(A_{si}, M_{si}), (R_1, A_{si}), (R_2, A_{si}), \ldots, (R_N, A_{si})\} \\ \cup \{(M_{si}, A_{ak}), (A_{ak}, M_{r1}), (A_{ak}, M_{r2}), \ldots, (A_{ak}, M_{rN})\} \\ \cup \{(M_{r1}, A_{r1}), (M_{r2}, A_{r2}), .., (M_{rN}, A_{rN}), \\ (A_{r1}, R_1), (A_{r2}, R_2), .., (A_{rN}, R_N)\} \\ M_0 = \left[\underbrace{0, \ldots, 0}_{N+1}, \underbrace{1, \ldots, 1}_{N} \right]. \end{cases}$$

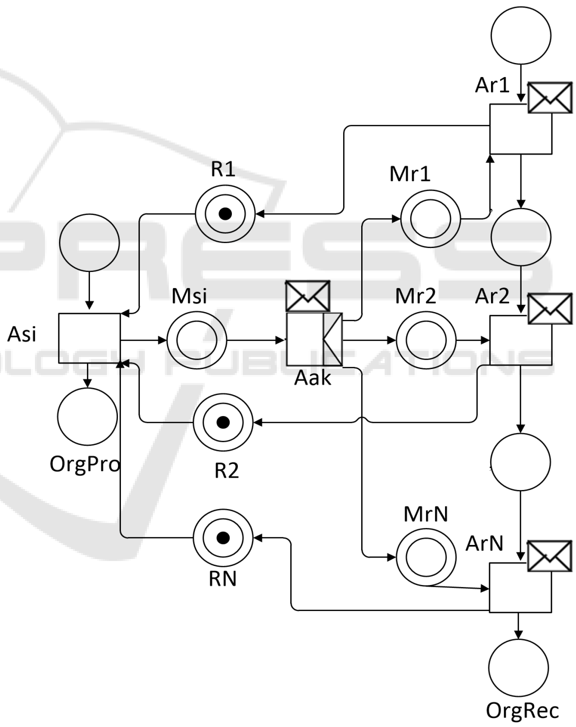

Figure 5: SEME adaptation pattern $(\text{SEME}_{\text{adapt}})$.

The adapter has only a single entry sending activity A_{si} got by place M_{si}. The message getting from M_{si} will exit to N receiving activities. Single to Multiple adaptations from the entry to the exits is performed by A_{ak}. The adapter splits one message M_{si} to N messages: $M_{r1}, M_{r2}, \ldots, \text{M}_{r\text{N}}$. $\text{SEME}_{\text{adapt}}$ pattern starts a working cycle upon the run of A_{si} and completes the cycle upon the run of the N receiving activities $\{A_{\text{r1}}, \text{A}_{\text{r2}}, \ldots, \text{A}_{\text{rN}}\}$. Accordingly, the run of A_{si} consumes the resource tokens in places

$\{R_1, R_2, \ldots, R_N\}$ and the occurrence of the N receiving activities $\{A_{r1}, A_{r2}, \ldots, A_{rN}\}$ returns all tokens to Triggers $\{R_1, R_2, \ldots, R_N\}$.

5 RELATED WORK

Patterns have been introduced in the workflow area for analyzing the expressiveness of business processes (Van der Aalst et al., 2003). In (Weber et al., 2008), Weber *et al.* proposed a set of 18 change workflow patterns and 7 change support features to enhance flexibility in the context of process-aware information systems (PAIS). However, proposed patterns do not support advanced change scenarios (e.g., adapting data when changing control).

In (Leshob et al., 2017), Leshob *et al.* introduced a pattern-based approach to adapt generic cross-organizational processes according to the organizations' specific needs. While this approach adapts business process views (i.e., dynamic, functional, organizational and informational) and insures their consistency, it does not specialize/adapt information systems that support them.

In (Fdhila et al., 2015), Fdhila *et al.* presented an approach to enable process adaptation in cross-organizational processes. Authors proposed a generic change propagation approach to adapt all partners' processes when a partner adapts its private process. However, proposed algorithms consider the application of one change at a time where, in practical scenarios, several change operations might be applied in a combined manner.

In (Ben et al., 2015), authors proposed six adaptation patterns for modeling collaborative processes. However proposed patterns lack formalization, to obtain unambiguous pattern definitions which will allow their implementation in collaborative process support systems.

6 CONCLUSION

Modeling cross-organizational business processes is complex and requires that designers have extensive experience. Indeed, modeling such processes require putting together a collection of private business processes from multiple partners that are often geographically dispersed. The challenge is even bigger when private processes are incompatible or have systems that support them such as PAIS.

The purpose of this ongoing work is to assist organizations in the process of modeling cross-organizational processes. To this end, we identified a set of six adaptation patterns that resolve incompatibilities when integrating organizations' processes.

Although this work is still at an early stage, this paper establishes guidelines to advance our long-term research project which consists of 1) analyzing private processes, 2) identifying incompatibilities, 3) selecting process adaptation patterns, and 4) constructing cross-organizational processes (see Section 2) using an iterative process.

REFERENCES

Ben, S., Chaabane, M., Bouaziz, R., and Andonoff, E. (2015). Flexibility of collaborative processes using versions and adaptation patterns. In *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*, pages 400–411. IEEE.

Fdhila, W., Indiono, C., Rinderle-Ma, S., and Reichert, M. (2015). Dealing with change in process choreographies: Design and implementation of propagation algorithms. *Information systems*, 49:1–24.

Leshob, A., Mili, H., Gonzalez-Huerta, J., and Boubaker, A. (2017). A value-oriented approach to business process specialization: Principles, proof-of-concept, and validation. *Journal of Systems and Software*, 127:120–149.

Li, X., Fan, Y., Madnick, S., and Sheng, Q. Z. (2010). A pattern-based approach to protocol mediation for web services composition. *Information and Software Technology*, 52(3):304–323.

Van der Aalst, W. M. (1998). The application of petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66.

Van der Aalst, W. M., Ter Hofstede, A. H., Kiepuszewski, B., and Barros, A. P. (2003). Workflow patterns. *Distributed and parallel databases*, 14(1):5–51.

Wang, K. W., Dumas, M., and Ouyang, C. (2007). Specification and analysis of adapters for conversational services. *eprints.qut.edu.au.*

Weber, B., Reichert, M., and Rinderle-Ma, S. (2008). Change patterns and change support features enhancing flexibility in process-aware information systems. *Data & knowledge engineering*, 66(3):438–466.

Zeng, Q., Sun, S. X., Duan, H., Liu, C., and Wang, H. (2013). Cross-organizational collaborative workflow mining from a multi-source log. *Decision Support Systems*, 54(3):1280–1301.