ISSN 2047-3338

Business Process Collaboration: A Fragment-based Approach using Connectors

Mounira Zerari
New Technology of Information and Communication Faculty, Mentouri2 University, Constantine, Algeria
zerari.mounirabis@yahoo.fr

***Abstract*—Contemporary enterprises and their business processes are becoming more dynamic, distributed and complex. However, traditional information systems struggle with requirements to provide interoperability due to the dynamic and heterogeneous nature of the modern business environment. Modularity concern is a common strategy to a lower application complexity: it results in modules that are easy to maintain, to adapt, and to replace. In this paper, we propose a new approach based on process fragments. We consider a process fragment as an independent part of process. Each activity is encapsulated in an entity named artifact. These artifacts have an interface, a function and manual instruction described with very expressive formal languages that manage high level concepts. On the other hand, the autonomy and heterogeneity of business partners require approaches to describe relations, interactions and collaboration between concerns and to reconstruct the full business process. Thus, this approach investigates this issue by adopting the notion of connectors in order to increase the reusability and to adapt the notion of artifact to the context of partners.**

***Index Terms*—Business Process Fragment, Artifact, Interoperability, Business Process Collaboration and Business Process Interaction**

I. Introduction

BUSINESS Process Management Systems have been regarded as one of the main types of the next generation of information systems. Business processes need to be able to adapt to changes in open business systems environments, such as arising in web applications and healthcare systems.

The full integration of all processes within a supply network is an upcoming challenge that has to be addressed by network partners and solution providers. The static integration of inter-enterprise processes as common in past years can no longer meet the new requirements of customer orientation, flexibility and dynamics of cooperation. A dynamic integration of processes becomes mandatory, which is obviously much harder to achieve [1]. Because of the deep development of economic globalization, enterprises tend to collaborate closely with others to keep their competitiveness by using other enterprises' valuable services as their own services' complement and make their own services potentially available to others. In addition, enterprises have to integrate their resources to provide fast and efficient responses, i.e., they realize business flexibility. In order to resolve the above problems, collaborative business process is widely studied in the scientific domain and industrial domain. Collaborative business process aims to define business collaboration requirements, not only between different enterprises but also between different departments of a same enterprise. Evidently, collaborative business flexibility includes inter- and intra- organizational workflows [2].

Workflows are used to specify business processes, at an abstract level. Each task in a workflow represents a unit of activity and can be completed by using a service. Several research work on Business Process Management (BPM) report that it is paired well with Service Oriented Architecture (SOA) to produce flexible business software solutions [3]. BPMs are gaining momentum thanks to the emergence of SOA. We consider that business processes are the workflow models viewed from the business domain. So, a workflow is a connection of entities encapsulating activities, or tasks called artifact. Each artifact represents a unit of work that contributes to a wider goal [4]. Viewing a workflow as a set of fragments permits to lower workflow complexity, fragments are easy to maintain, to adapt, and to replace. However, Business Process Integration (BPI) and interaction between several business fragments have to be considered. Indeed, the community is still debating the issues of enterprises collaboration at the business process level.

Our research focuses on the business process collaboration in a flexible way. We have already proposed a context-aware process mining framework for business process flexibility (underspecification business process flexibility) [5]. We show how the notion of artifact increases flexibility in an intra-organizational way. In this paper we focus on collaborative business process fragments in the service architecture environment. We propose a construction approach that specifies how the integration and the interaction of one process fragment with another process fragment is achieved

in a dynamic way. Our objective is to exploit the formal process algebra language used in manual instruction of an artifact for specifying those interactions [6].

The remainder of this paper is structured as follows: Section 2 presents some background material on workflows and artifacts. Sections 3 describe the overall proposed approach. The description of connectors' specification is given in section 4. Section 5 presents an Example to show how the approach can be exploited. The paper is rounded of with related work in section 6 and a conclusion in section 7.

II. BACKGROUND

The goal of this section is to show that business processes are indeed connected fragments that interact and need to be integrated. In a previous work [5], we encapsulate business process fragments into entities or components. Each entity possesses an interface for communication and a manual or operating instructions. Based on information collected from the runtime, helped by techniques of process mining, a decision will be taken to execute one of the process fragments (Figure 1).

Figure 1. Process fragments

We define a BP (Business Process) using artifacts formally as follows:

$$BP =< AV, AR, AE, AS, AT, RS >$$

Where:

AV: a set of activities composed the BP, AV_i is an activity;

AR: a set of roles that operate in the BP AR_i is a role;

AE: a set of input parameters, AE_i: input parameters of AV_i;

AS: a set of output parameters, AS_i output parameters of AV_i ;

AT: a set of transition condition AT_i denote AV_i transition;

RS: a set of resources invoked in BP execution;

An artifact that encapsulates an activity is defined as a quadruple:

$A =< F, Att, UI, M >$

- F: It is the function that describes the service proposed by the entity. It is the aim of activity.
- Att: It concerns the attributes. They are data and parameters internal to artifact. Exhibited by artifact in order to represent dynamic properties. Those attributes are used by other components to be configuring.
- UI (User Interface): It is a specification describing the behavior of the artifact. It is defined as a set of operations. Two kinds of operations: execution of an action and the perception of the end of an action:

Action = {start (input data), pause, restart, stop, ...}
Perception = {finished (result), failure (errorinfo)}.

Action and perception defined above are basic definitions. We will describe others according to the definition of business process.

- M: It is a manual represented by a set of formal instructions which describe the manner that the artifact is executed. Those instructions are described by a formalism based on process algebra. Nevertheless a short description of principles can be presented. An Instruction is defined as follows:

$$I ::= 0|!\alpha|?\pi|I; I|I + I|(I||I)|D(t1, ..., tn)$$

I can be an atomic Instruction: Behavior ZERO (0), execution of the action $!\alpha$ and the perception of the end of the action $?\pi$. I may also be structured using different operators: ";" for the sequential composition, "+" for the choice and "||" for parallel composition. The concept of recursive can be assured by the invocation of D (t1, . . . ,tn) of another basic instruction where D is the instruction invoked and (t1, . . . ,tn) is a list of its parameters.

A major prerequisite for a dynamic integration is the reliability and flexibility of the involved processes. Although companies will work together. Indeed, as intra-organizational workflows have been studied for a long time and have lots of research results, so in this paper, collaborative business processes focus on inter-organizational workflows.

Furthermore, even integrated business processes are subject to change, i.e., a company commits itself to providing a process with a certain input and result, but still wants to retain the freedom to internally reorganize and optimize its processes as necessary. So, a public business process is the aggregation of the private processes and/or web services participating in it.

So, there are many views of a BPM lifecycle. This paper prefers the BPM lifecycle including the phases:"definition", "modeling", "deployment", "execution", "monitoring", "analyzing" and "optimization" and at last back to "definition" [7]. Furthermore, considering business process collaboration and interoperability, this paper is concerned about the phases"deployment" and"execution". The need of interoperability was initially a design time concern; while today's dynamic environment execution requires on-the-fly collaboration.

In our context, as we mentioned above, we consider a business process as a set of artifacts. Each one encapsulates an activity.

In order to achieve interoperability and collaboration, dynamic business process integration will be achieved in an open environment by observing the execution environment. The interface for each artifact permits the observation.

Information attached to own artifact is available through the manual. This last can be used to reason and decision for interaction can be taken. The notion of artifact permits a loosely coupled description of business processes.

In the following, we propose a new approach inspired from emergent connector's ADLs (Architecture Description Languages) and Coordination Languages (CLs) for specifying the collaboration between several business process fragments that are represented as artifacts. They are, in our opinion, good candidates to describe such interaction between fragments in an abstraction and adaptive ways.

III. A FRAGMENT BASED APPROACH USING CONNECTORS

Constructing a public business process from private fragments is known to be a difficult task, in particular in a dynamic context and where fragments have not been designed altogether from the beginning. Therefore, the adaptation connectors are needed.

A. *The notion of connectors*

Connectors are architectural building blocks used to model interactions among components and rules that govern those interactions in ADLs context. Medvidovic and Taylor [8] state that connectors are a particular kind of component used to model interactions which, however, may not correspond to a compilation unit in the implemented system. In our context, we consider the connectors as an abstract glue to express interactions, which provides a good balance between formality and expressiveness without any sacrifice. Instead of using a rigid encoding of the integration of the private business process parts, we choose to use separated glue i.e. external to the business process fragments and put in an external structuring view.

B. *Coordination and cooperation of business processes*

Collaborative business processes can be considered as two kinds: the coordination of business processes and the cooperation of other ones [2].

A coordination of business process: is composed of the activities some of which take place between enterprises, but the process execution is owned and controlled by only one enterprise. Figure 2 shows an example of coordination business process, in which Enterprise A uses services provided by Enterprises B and C.

A cooperation of business process: is composed of the activities some of which take place between enterprises. The process execution is owned and controlled by enterprises, but each enterprise can only control the execution of its own activities. There are two expression views for cooperation business process: a centralized view and a distributed view.

In the first one, the activities of all enterprises in cooperation are modeled in one business process. For example, in figure 3, the activities of enterprises A and B are in the same process.

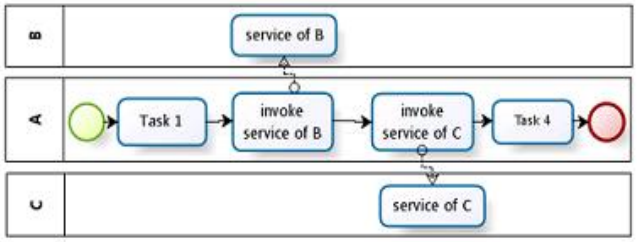

Figure 2. Example of coordination business process[2].

Figure 3. Example for centralized view of Cooperation Business Process[2].

A cooperation of business process: is composed of the activities some of which take place between enterprises. The process execution is owned and controlled by enterprises, but each enterprise can only control the execution of its own activities. There are two expression views for cooperation business process: a centralized view and a distributed view.

In the first one, the activities of all enterprises in cooperation are modeled in one business process. For example, in figure 3, the activities of enterprises A and B are in the same process.

In the distributed view, the activities of each enterprise are included in its own business process, and the collaboration between enterprises is expressed by their message exchange. For example, in figure 4, the activities of A or B are included in their own processes, and their collaboration is the message exchange. This view was proposed by [9].

C. *Representing the fragment by the artifact*

In this approach, we consider two types of business processes, the private processes and the public ones. The first type is considered as the set of processes of the company itself and they are managed in an autonomous way.

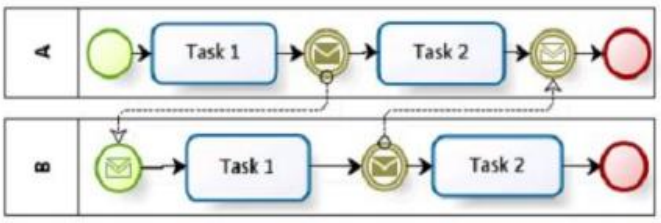

Figure 4. Example of distributed view of cooperation business Process[2].

Private processes are supported within companies using traditional Workflow Management Systems. These systems were intended to serve local needs and can use a service

proposed by an external enterprise (coordination of business processes).

In the other hand, public processes span organizational boundaries. They belong to the companies involved in a B2B relationship and have to be agreed and jointly managed by the partners (Figure 5).

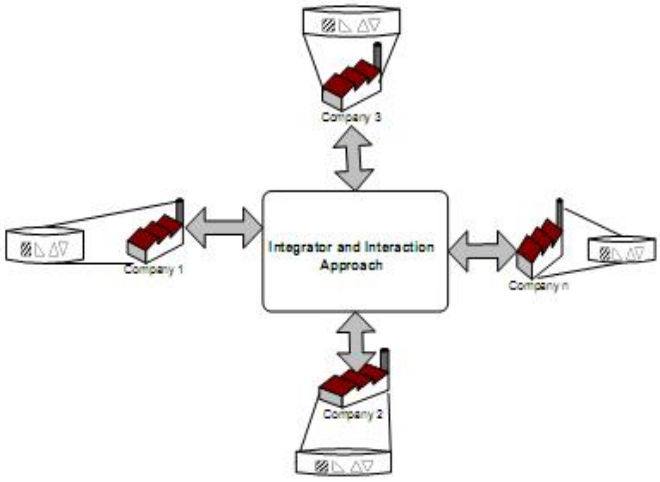

Figure 5. A global View of public and private business process.

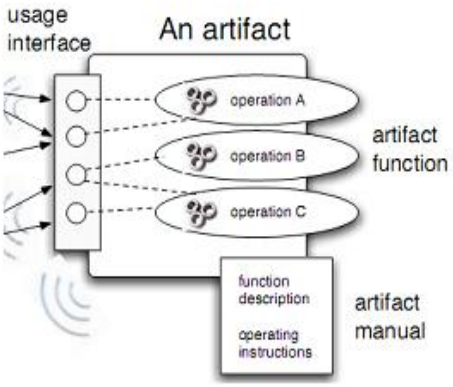

Figure 6. The representation of a business fragment as an artifact.

In this case, it will be a cooperation business process. Each private business process is a set of artifacts that are interconnected in order to achieve a local goal. However, the use of local artifacts from different companies to compose public business processes can be benefic. The modularity, the formal aspects and reuse of artifacts are features that permit increasing flexibility, adaptability and dynamic B2B integration scenarios.

In an earlier work [5], we have considered an activity (task) of a business process as an entity encapsulated by an artifact (Figure 6) in order to achieve underspecification business process flexibility. The execution context of a business process in run-time is considered to choose the appropriate artifact. In the B2B integration context typically involve distributed business processes that are autonomous to some degree. Companies participating in this scenario publish and implement a public process. The application integration based on public processes is not a new approach.

Our approach aims to exploit modularity achieved by artifacts to facilitate B2B integration. In order to determine and reason about different options to solve the heterogeneity and variability of protocol's companies, Our approach is inspired for modeling our system by concepts of software architectures [10] (Figure 7).

For using an artifact, the manual instruction is expressed in a formal language:

Simple staffware: = ((!registration; ?end registration);
(!send questionnaire; ?end send questionnaire);
!receive questionnaire; ?end receive questionnaire));
(!evaluate; ?end evaluate); simple staffware.

The communication with the artifact is achieved through the interface in order to take decisions (Table I).

Table I
USING ACTION AND PERCEPTION TO EXECUTE AN ARTIFACT.

Action	Precondition	Perception	Effect
Registration	¬Registration	EndRegistration	Registration

The contribution of this work is based on the consideration of artifacts as software building blocks provided with one or multiple communication interfaces. Each one corresponds to a communication port. Then, an artifact has a set of ports that correspond to the all points of interaction with the outside world. Ports can also be seen as access points to the artifact. The description of an artifact (using the process algebra) may include the specification of a global behavior that coordinates all its ports.

Connectors, for their part, model the interactions between artifacts by defining the rules governing these interactions. The connectors have several communication interfaces or roles.

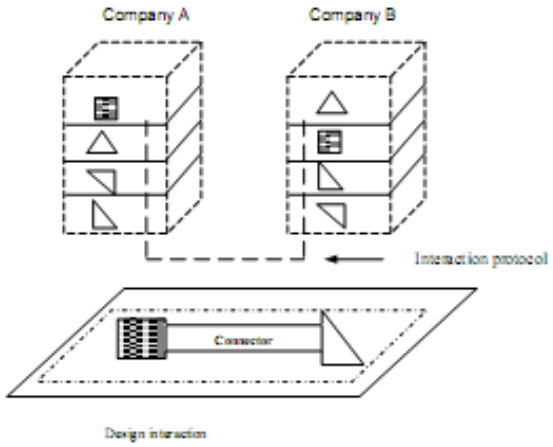

Figure 7. An overview of the proposed approach.

Each role defines the expected behavior of participating in the interaction. The specification of a connector also includes

the definition of a coordination protocol (called glue) that specifies the coordination of the different roles of the connector. The behavior of the roles and glue are expressed as for the artifacts, using a process language.

IV. CONNECTORS SPECIFICATION

We need to address a connection among artifacts for inter-enterprise interaction (conversation, also known as choreography, in the context of Web services). We consider a connection, in terms of software architecture, as a specification that relates to the one of interaction protocols associated with component ports to ensure conformance with connector role. Hence, from the standpoint of roles and connector, we note that in process algebra: a → P means Action Prefix and a → P| | b −→ P means a choice. The interconnection definition between artifacts is modeled as:

// Connectors specification
Role $Artifact_1$= | i ∈ [1,.., n](glue$_i$ → R_{1i}). $Artifact_{ii}$ ∈ [1,..,n] = Initial specification of the $artifact_1$ given by the connector.
Connector$_i$ | reset ⟶ R_1

Role $Artifact_2$ = | i ∈ [1,.., n] (glue$_i$ → R_{2i}),
$Artifact_{2i}$ i ∈ [1,..,n] = Initial specification of the $artifact_1$ given by the connector
connector$_i$ | reset ⟶ R_2
$Glue_1$ = The glue specification that coordinates $artifact_{1i}$ and $artifact_{2i}$.
// Set of events initiated or observed

set $I1_{i,i\in[1n]}$ = Set of events initiated from Role $Artifact_1$
set $O1_{i,i\in[1n]}$ = Set of events observed from Role $Artifact_1$
set $I2_{k,k\in[1m]}$ = Set of events initiated from Role $Artifact_2$
set $O2_{k,k\in[1m]}$ = Set of events observed from Role $Artifact_2$

Yet, our approach is based on process algebra to reason on partner heterogeneity in order to be independent of their underlying technology (publishing public business process). The introduced systems achieve interoperability among existing protocol without modifying the application.

V. EXAMPLE

In order to illustrate the exploitation of our approach, we consider a hospital environment (health-care systems). Let us consider the flow of the admission of patients in a hospital as a workflow process. The activities in such a workflow include all kinds of treatments, operations, diagnostic, tests, etc. Each activity is represented by an artifact. When a patient is admitted in a hospital, it's important to have his medical history (Figure 8). For simplicity, we focus on the interactions between activity (artifact) "Triage" in hospital A (Enterprise A) and "updated patient history" in hospital B (Enterprise B) (Figure 8). The specification of the interaction is represented as follow:

FileSharingTriage = (req.searchFile → P1),
P1 = (req.downloadFile → P1
| req.downloadComment → P1 |terminate →END).
FileSharingUpdate = (req.uploadFile → FileSharingUp-date
| terminate → END).
FileSharingUpdate = (prov.uploadPhoto → FileSharingUpdate
| prov.searchFile →FileSharingUpdate
|prov.downloadFile → FileSharingUpdate
|prov.commentFile
|FileSharingUpdate
| terminate → END)
Glue-example = c.open →s.open, c.request → s.response → Glue-example, s.response → c.response → Glue-example, c.close → s.close → Glue-example
set Triage FileSharing Actions ={searchFile, downloadFile, downloadComment,downloadradio}
set Triage FileSharing Perception = {finedFile, down-loadedFile,downloadedComment, downloadedradio}
set Update FileSharing Actions = {uploadFile}
set Update FileSharing Perception = {uploadedFile}

From the above, it is quite trivial to show the interaction between the two parts of business processes. Hence, they can be dynamically composed, while ensuring correct system behavior with respect to the provided properties. However, this mechanism needs to be realized by a middleware support.

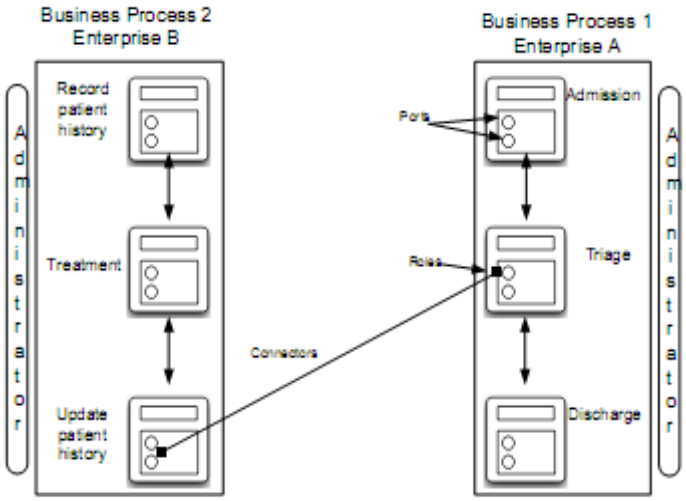

Figure 8. Admission patient distributed on two hospitals

VI. RELATED WORK

Component reuse is defined as the process of implementing or updating information systems using existing assets, and the

ability of reusable components to be varied and appropriate to different designers and re-users needs is a key property in reusable components development. This process has been supported by research contributions from various fields [11]. As well as the notion of public process and private process is not new.

Benmerzoug et al [12] propose an approach where the business processes integration is modeled using AUML and specified using BPEL4WS. The authors propose translating rules for the conversation of an interaction protocol given in AUML to CPN. Unfortunately, no procedures were provided that guide the conversion of an interaction protocol given in AUML to Petri net representations.

In [3], they consider that a workflow should have enough information attached to it to run successfully on its own.

[2] Uses MDA (Model Driven Architecture) which is an approach to using models in software development. It divides models into three levels and uses model transformation to generate low-level models from high-level models. MDA aims at improving automation degree of software development.

Compared with the related work, our approach allows us to provide a clear separation of inter-enterprise collaboration management and local business process management, to make full use of existing workflow system components, to support both public processes and private business processes. Another advantage of our approach is the adaptability and flexibility that formal languages provide to the interconnection procedure. Since applications integration is often viewed as a hierarchy of different local systems and services, the notion of connector achieves interoperability and allows overcoming the heterogeneity of public business partners. The activity of a process, in our approach, is encapsulated in an entity considered as a component. This component is largely inspired by artifact in coordination in MAS (Multi Agent System) [6]. This approach covers the coordination business processes and cooperation business processes.

VII. CONCLUSION

After describing the notion of artifact and its advantages, this paper focuses on the exploitation of software middleware principles to formalism the coollaboration of business processes.

The interactions are considered in the context of BPMSs. In particular we consider systems that are described by a workflow that has a local goal and a global one. The tasks in the workflow are considered as artifacts. From this standpoint, we propose a generic approach that permits the exploitation of modularity, the autonomous and the independence of artifacts to support the B2B interactions. We define the interaction as a port in a formal way that supports heterogeneity in an independent way.

This approach supports the flexibility and adaptability integration in a dynamic way. It covers also the business process coordination and cooperation.

Future work includes the formalization of the port interaction which is the development of a formal semantics for the artifact part. This last one will allow to extend the reasoning techniques for adaptability for changes.

REFERENCES

[1] J. Koehler, G. Tirenni, and S. Kumaran, “From business process model to consistent implementation: A case for formal verification methods,” in 6th International Enterprise Distributed Object Computing Conference (EDOC 2002), 17-20 September 2002, Lausanne, Switzerland, Proceedings. IEEE Computer Society, 2002, pp. 96–106.

[2] H. Liu, Y. Lembaret, D. Clin, and J. P. Bourey, “Comparison between collaborative business process tools,” in Proceedings of the Fifth IEEE International Conference on Research Challenges in Information Science, RCIS, IEEE, Ed., Gosier, Guadeloupe, France, 19-21 May 2011, pp. 1–6.

[3] S. Gorton and S. Reiff-Marganiec, “Towards feature interactions in business processes,” in International Conference on Feature Interactions in Software and Communication Systems,ICFI 2007, 3-5 September 2007, Grenoble, France, 2007, pp. 99–113.

[4] M. Zerari and M. Boufaida, “An artifact-based architecture for a better flexibility of business processes,” in ICEIS 2010, 2010, pp. 359–365.

[5] “Context-aware process mining framework for business process flexibility,” in ACM iiWAS 2010, Paris, France, November 8-10, 2010, 2010, pp. 419–424.

[6] C. Dinont, P. Mathieu, E. Druon, and P. Taillibert, “Artifacts for time-aware agents,” in AAMAS, 2006, pp. 593–600.

[7] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, “Business process management: A survey,” in Business Process Management, International Conference, Eindhoven, The Netherlands, June 26-27,, ser. Lecture Notes in Computer Science, vol. 2678. Springer, 2003, pp. 1–12.

[8] Nenad Medvidovic and R. N. Taylor, “A classification and comparison framework for software architecture description languages.” in IEEE Transactions on Software Engineering,, vol. 26, no. 1, 2000, pp. 70–93.

[9] W. M. P. van der Aalst and M. Weske, “The p2p approach to interorganizational workflows,” in Advanced Information Systems Engineering, 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, ser. Lecture Notes in Computer Science, vol. 2068. Springer, 2001, pp. 140–156.

[10] V. Issarny, A. Bennaceur, and Y.-D. Bromberg, “Middlewarelayer Connector Synthesis: Beyond State of the Art in Middleware Interoperability,” in 11th International School onFormal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems, M. Bernardo and V. Issarny, Eds. Springer, 2011. [Online]. Available: http://hal.inria.fr/inria00586630/en.

[11] R. Saidi, A. Front, D. Rieu, M. Fredj, and S. Mouline, “From a business component to a functional component using a multi-view variability modelling,” in MoDISE-EUS’08, the International Workshop on Model Driven Information Systems Engineering, Montpellier, France, June 16-17, 2008. CEUR-WS.org, 2008, pp. 34–45.

[12] D. Benmerzoug, F. Kordon, and M. Boufa¨ ıda, “Formalization and verification of interaction protocols for business process integration: Petri net approach,” Int. J. Simulation and Process Modelling, vol. 4, no. 3/4, pp. 195–204, 2008.