

Enhancing Network Collaboration in SOA Services Composition via Standard Business Processes Catalogues

Roque Bezerra, Maiara Cancian, Ricardo Rabelo

► **To cite this version:**

Roque Bezerra, Maiara Cancian, Ricardo Rabelo. Enhancing Network Collaboration in SOA Services Composition via Standard Business Processes Catalogues. 18th Working Conference on Virtual Enterprises (PROVE), Sep 2017, Vicenza, Italy. pp.421-431, 10.1007/978-3-319-65151-4_38 . hal-01674894

HAL Id: hal-01674894

<https://hal.inria.fr/hal-01674894>

Submitted on 3 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Enhancing Network Collaboration in SOA Services Composition via Standard Business Processes Catalogues

Roque O. Bezerra¹, Maiara H. Cancian² and Ricardo J. Rabelo¹

¹ Department of Automation and Systems Engineering
Federal University of Santa Catarina
Florianopolis (SC) Brazil
{roque.bezera; ricardo.rabelo}@ufsc.br

² Estácio Florianopolis
Rodovia SC401 km 01 – Florianopolis (SC) Brazil
maiara.cancian@estacio.br

Abstract. Resources sharing between members are a key issue in Collaborative Networked Organizations (CNO). In the software services sector most of companies develops their services by their own and stores them at their local silos without sharing them with other partners. However, the development of services-based applications can be very complex and costly, which is a difficult issue to overcome as most of IT companies are SMEs. In this line this paper proposes a digital catalogue environment to leverage services sharing and larger reuse between CNO members. It is strongly grounded on standard business processes models that would be adopted by all the involved members. The catalogue acts as a collaborative environment that logically embraces all the public services made available by the CNO members, enabling software services developers to compose new services-based applications more agilely. A prototype has been implemented and its results are presented and discussed.

Keywords: Business Process, Catalogue, Repository, SOA and UBL.

1 Introduction

Likewise in other sectors, SMEs (Small and Medium Sized Enterprises) from the software sector have been increasingly pushed to adopt advanced IT and more sustainable business models to stay competitive in the market [1, 2]. However, they use to be very limited in their capacity to engage the required assets for that. Collaborative Networked Organizations (CNO) has emerged as a powerful strategy for SMEs to overcome such limitations [3].

In this sector, *SOA* (Service Oriented Architecture) and services-based principles have been gradually adopted by software companies to more agilely create new solutions that are better aligned to their business strategy as well as to foster more advanced business models. Recent advances on IT have grounded the emergence of new models, e.g. the ones based on the vision of larger scale provision and offering of software services by pervasive providers from digital ecosystems, that are distributed over the Internet and that can be accessed on demand, from everywhere, anytime [4].

Working in a CNO can leverage many competitive advantages. For example, the sharing of resources [5], including software active assets [4]. Such collaboration dimension can help software services providers to save development costs, share

losses and offer more value to customers [6]. Empirical observations show that, however, such providers use to keep their (*web-services*-like software) deployed exclusively at their local silos to both support their own Business Processes (BP) and to attend their customers' process/systems needs [7]. Operationally, this prevents providers from increasing the potential of services' reuse and ROI (*Return of Investment*) as services are not shared among companies. Strategically, this prevents them from the benefits of getting involved in larger value chains [4]. This limitation gets even bigger as providers and customers use to adopt their own and proprietary BP models [7], which ends up requiring made-to-fit and hence more expensive IT solutions. This, in turn, use to create vendors dependency and technology lock-in [7].

An approach to face this issue is via BP catalogues and repositories. Yan et al. [8] have highlighted the importance of having catalogues to organize, manage and handling BPs repositories and their life cycle in an organization. Nurmilaakso [9] has pointed out the potential of a larger adoption of standard BPs by organizations in terms of processes interoperability and reuse. Actually, a number of BP models have been proposed since decades (e.g. *ISO 19440*, *ISA-95*, *EDIFACT*, *ebXML*, *Rosettanet* and *UBL*), although with an emphasis on interoperability and BP modeling [7]. Many works also have been addressing BP management and BP:SOA layers integration to enhance business agility and IT alignment (e.g. [10]).

This paper investigates the hypothesis that such issues could be mitigated if CNO members adopt catalogues and repositories based on open and standard BP reference models to boost their collaboration so as to increase their competitiveness.

Applying the Design Science methodology [11], a standard-based BP digital catalogue environment was built to experimentally and mostly qualitatively verify the general feasibility of its use as an open and "unified" collaborative platform to create SOA/services-based applications. This artifact was used to create a scenario where reference BP models would be largely adopted by both software providers (when they develop their software services) and by their customers (when they specify their internal business processes). This would create a global logical view over all the software services (repositories) developed by the providers (and even by some customers) members of a CNO so that new SOA/services-based applications would be composed in a more agile and BP-coherent ways.

Although the model is flexible to deal with other BP models, UBL (*Universal Business Language*) [12], from OASIS, has been chosen for this proof of concepts regarding it is open, free and comprehensive.

This paper is organized as follows. Section 1 has introduced the problem and paper's goal. Section 2 summarizes the review of related works and identifies the scientific contribution of this research. These two sections represent the expression of awareness of the problem and the initial basis for the intended artifacts' design in the Design Science methodology. Section 3 describes the developed catalogue's rationale, its prototype and experimental results, representing the requirements, the artifact itself and the performance measurement steps in Design Science. Finally, section 4 presents some conclusions, representing the achieved results step.

2 Literature Review

As the result of a literature review it was identified that a number of authors have made important contributions on some of the issues pointed out in section 1. They have inspired this paper's authors in the design of the envisaged catalogue

environment. For example, Cancian *et al.* [4] have identified the list of BPs that software service providers have to support in the collaboration life cycle (including services provision and support) considering aspects like partners' and services' certifications, governance and trustworthiness. Perin *et al.* [10] have developed a dynamic services discovery environment that works over disparate UDDI-based services repositories considering BPs' context and QoS to better support BPM-SOA integration. Using different IT supporting approaches, Camarinha-Matos *et al.* [13] and Obidallah [14] have developed platforms to identify and to better link BPs' needs to partners' competences and software services within a CNO. Rabelo *et al.* [15] have developed an open and plug & play SOA/services-based ICT platform to support dynamic offering of software services by CNO members and their link to collaborative BPs, including the organization of services within a so-called federation. Pinheiro *et al.* [16] have used a grid computing platform to support the sharing of computing resources (memory and storage) between CNO members.

Related to BP catalogues, it was observed that the existing ones are essentially repository management systems of proprietary BPs' models, including BP mining in some tools [8]. Eighteen catalogues & repositories were found out in the search, but only six were considered as relevant and compared with the one being proposed in this work regarding the tackled scenario (Table 1). This comparison, however, does not aim at stating which one is the "best", or that the one developed is a full-fledged environment. Instead, it aims at highlighting the main differences and commonalities with the proposed BP catalogue when supporting that scenario.

The evaluated catalogues were: *MIT Process Handbook* [17], *Semantic Business Process Repository (SBPR)* [18], *IBM BPEL Repository* [19], *RepoX* [20], *Oryx* [21] and *APROMORE* [22]. The requirements' subset used in Table 1 came from the surveys [7,8], and the 'CNO support' aspect was added by the authors as it was not covered by these two surveys. The classification *yes* does not mean the full support for the elicited requirements, but rather at least some basic support.

Table 1. Basic comparison among BP catalogues

| Requirements | Process Handbook | SBPR | IBM BPEL | RepoX | Oryx | APRO-MORE | Proposed Model |
|-----------------------------------|------------------|------|----------|-------|------|-----------|----------------|
| Reusable BPs | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| BP Language Independence | Yes | No | No | No | Yes | Yes | Yes |
| Flexibility for BPs extensions | No | Yes | Yes | Yes | No | No | Yes |
| Semantics support | No | Yes | No | No | No | No | Yes |
| Non-proprietary BP model | No | No | No | No | No | No | Yes |
| Non-proprietary modeling language | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| SOA integration | No | No | Yes | No | No | Yes | Yes |
| BPM editor integration | No | No | Yes | Yes | Yes | No | Yes |
| Open source | No | No | No | No | No | No | Yes |
| CNO support | No | No | No | No | No | No | Yes |

Regarding this paper's goals and as a summary of this general comparison, it was realized that none of those works or other similar found out in the literature have

proposed a BP catalog as a “common platform” to be used at CNO level enabling members to compose SOA/services-based software applications by means of massive reuse from a large-scale pool of shared services adopting open reference BP models.

About UBL, the *Universal Business Language*, it is a royalty-free library of standard XML business documents supporting commercial and logistical BPs for supply chains, such as procurement, purchasing, transport, logistics and intermodal freight management. UBL can act as a *lingua-franca* supporting disparate business applications and trading communities to exchange information using common format and terminology. It is modeled as XML schemas, which are modular, reusable and extensible, allowing its evolution and application to other domains [12].

3 Business Process Catalogue’s Model & Prototype

3.1 The Catalogue’s Rationale

The BP catalog has been conceived taking the requirements highlighted in the previous section into account, also considering the envisaged CNO scenario and current trends in BPM (Business Process Management) : SOA integration [7,8,23]. This includes the use of open standards in all parts of the entire environment in order to mitigate interoperability problems.

The catalog is basically represented by an environment through which right actors from a CNO of software services providers can more agilely generate new software applications that are compliant to the respective BPs’ UBL specifications when attending their customers. Customers would also adopt UBL. The applications themselves would be composed of a set of distributed software services.

Different services implementation models can co-exist. Equivalent services (from the functional point of view) can have different QoS levels, support different technologies- (e.g. web services, SOAP, etc.), billing- and access- models, multi-tenancy, etc., depending on each provider’s strategy and its internal development skills and practices. However, such equivalent services should adopt a common service’s interface (i.e. the same parameters for the service’s *WSDL*, in this case). Once published, they will act as the services’ references for further discovery, binding and invocation. There can be multiple equivalent services being provided by a sort of different providers for the same BP.

Providers are autonomous to decide which services will be shared within the federation also respecting contracted businesses and SLAs. The general management of such providers, their services’ governance, reputation, revenues distribution, CNO’s conventions, etc., as covered in [4], are subjects out of the scope of this work.

The to-be generated SOA/services-based application can be very complex in terms of e.g. interoperation, security and resilience capabilities. Besides that, services can be of different types, such as related to business, shop-floor, utilities, infrastructure, etc. A composed application via the catalog environment will rarely be ready to be deployed and executed. Actually a number of technical adjustments of many natures usually have to be made in real cases, besides the fact the, depending on the planned deployment and access models, non-software services can be required (e.g. manual integrations at the customers’ site, configuration of the eventual ESB in use, adjustments in the generated BPEL file due to BPMN limitations, helpdesk, etc.) [24].

A given SOA/services-based application can vary from customer to customer for the same BP depending on the required non-functional requirements, both at global

level (e.g. end-to-end QoS) and at individual services level (e.g. when a given system’s functionality should respect specific performance metrics). For example, an application to cope with the UBL *Invoicing* BP for some customer can be composed of a different number of services (and their versions and respective providers) than one for the same BP to attend other customer’s requirements.

3.2 The Catalogue’s Architecture and Prototype

The catalogue’s architecture can be seen as a partial instantiation of the BPM reference architecture proposed in [7]. It provides some support to design processes, handle applications’ repository and manage related services enactment [7].

The catalogue itself is a SOA application, being its modules implemented as services too. It is integrated to a BP editor (at the BPM level) and to a dynamic service discovery environment (at the SOA level), besides interacting with other modules. This global environment helps managers to design, reuse or modify existing services-based applications (developed following UBL), to discover and bind services from the CNO federation, to store the composed application, and to execute it when necessary. Figure 1 provides a general view of the catalogue environment’s architecture.

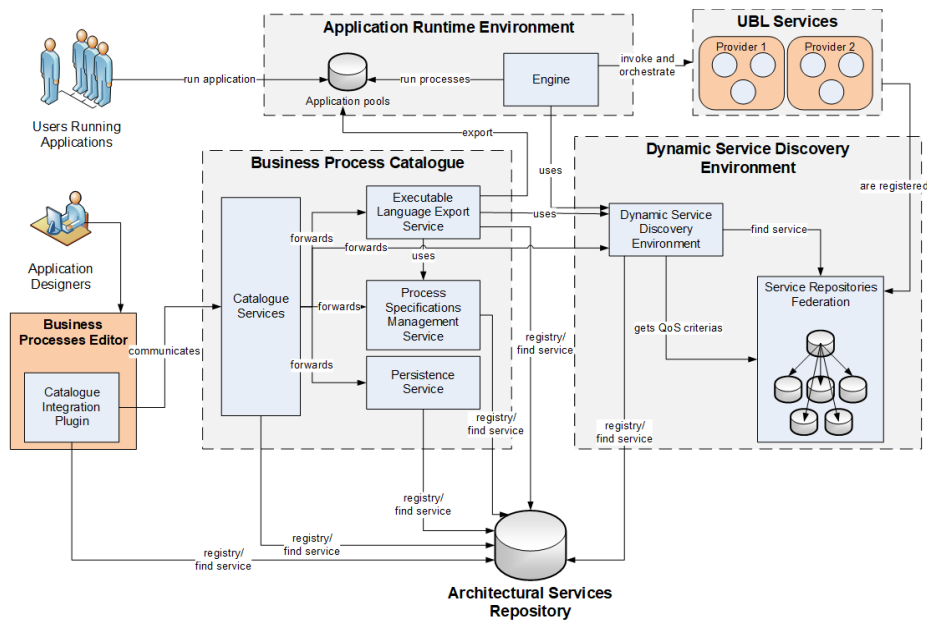


Fig. 1. Catalogue environment’s general architecture.

In general, the IT architect and/or the Process analyst from a given CNO starts interacting with the BPM editor looking for the (UBL) BP for which they want: to create a new (services-based) software application; to modify a previous one (to reuse it) already developed for e.g. an older BP’s version; or to bind another services to the application regarding new QoS requirements. The catalog provides the usual functionalities for this, as *edit, search, delete, merge, compose, monitor, save*, among

others. Users also specify the QoS for the application (out of 11 parameters, such as *performance* and *response time*), being further transformed into a SLA.

Figure 2 shows a case where the *Ordering* UBL process would be accessed from the BP catalogue aiming at generating a respective application. An auxiliary graphical interface allows assigning the required QoS. In the case this application had been already generated in the past (it would be stored in the catalogue BPEL’s repository) and it can be recovered for further refinements.

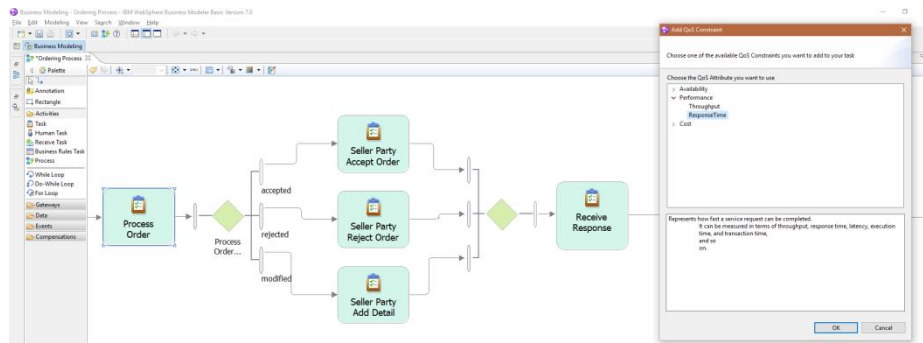


Fig. 2. Loading a given UBL process and QoS constraints.

Processes are modeled in BPMN and are further converted into BPEL files. Figure 3 shows an excerpt of the generated BPEL file.

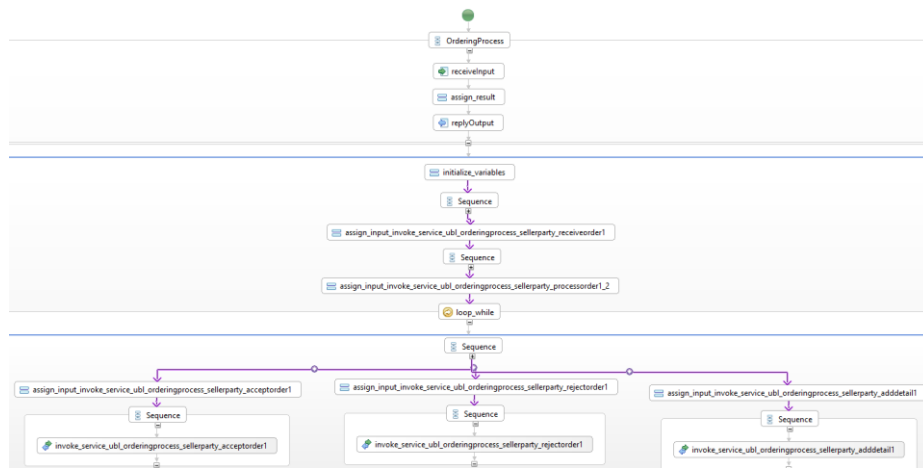


Fig. 3. Generated BPEL file.

The UBL repository can work over a single CNO member’s services repository or the so-called CNO federation, which creates a “cloud-like” view over the distributed providers’ repositories. The catalogue is a web-based application and uses an Internet browser as its front-end. The catalogue environment acts as a ‘unified’ platform through which the collaboration among consumers and providers is leveraged and

supported. Depending on the agreed deployment model, each member can have access to the federation and catalogues via such front-end.

One of the key elements of the catalogue environment is the discovery service [10], developed in the scope of this work. It supports the search, analysis, selection and final binding of the services (from the services federation) that fit best the set of functional and non-functional requirements of the application(s) under construction. The discovery service permanently checks in background the services' properties and availabilities, and maintains a ranking of up to five services for each BP's sub processes so as to always have a service ready to be invoked. This means that services binding are done dynamically instead of statically.

In the case none service can be discovered to match a given BP's requirement then the IT architect and/or Process analyst can or should relax some QoS constraint. In the worst case, if the situation persists then a new service should be developed.

As the BPEL execution is not triggered as soon as services are discovered, the involved actors can check the suggested (five) services for each BP's sub process and manually modify the ranking. This ranking is based on the QoS fitness range. Once everything is set up for the given BP then the respective BPEL file is stored into the BPEL catalogue repository for further execution.

Although the management of the federation is a subject out of the scope of this work, new services and repositories can be added or deleted from the environment. This dynamics is automatically handled by the discovery service as it always checks the registered services and available repositories.

Services and their interfaces should be previously and properly registered (following the SOA principles and the UBL specifications). Figure 4 shows a fragment of the code used to register a service related to the *OrderingProcess* UBL process, which has an activity called *placeOrder* that is performed by the *BuyerParty* actor. This is modeled in a UDDI information structure (metadata) devoted for that, called *tModel:uddi:ubl:services:ordering_orderingprocess_buyerparty_placeorder*. Every information (e.g., QoS attributes in this case) related to a given service has a tModel, and the UDDI supporting software has 'services' to access them. A generic *getServiceQoS()* method was implemented to get the desired tModels. In this example, the service's address (*endpoint*) is *http://examplecompany.com/services/ubl/orderingprocess/buyerParty/placeOrder*.

```
publishService(
    "UBLOrderingProcessBuyerPartyPlaceOrder",
    "UBL Payment -> Ordering Process -> Buyer Party -> Place Order",
    "http://examplecompany.com/services/ubl/orderingprocess/buyerParty/placeOrder",
    new TModelInstanceInfo[] {getTModelInstanceInfo("uddi:ubl:services:ordering_orderingprocess_buyerparty_placeorder")},
    getServiceQoS()
);
```

Fig. 4. Service registry in the UDDI

3.3 The Catalogue's Implementation Technologies and Deployment Environment

The catalogue was fully implemented in Java, using the *Eclipse* platform. The BPM editor has used the *IBM Websphere Business Modeler* tool and a plug-in was developed as a connector to support the implementation of the *UBL* specification. Process models are generated in *BPMN*. The whole SOA environment has adopted the *SCA* architecture (*Service Component Architecture*) and was executed in the *Apache*

Tuscany environment. The execution environment was supported by the *Intalio BPMS*, a suite that generates and uses *BPEL* (version 2.0 compliant), using the *Apache ODE*. *Hibernate/HSQLDB* was the database used to store the UBL BP models. Services are registered and stored using *jUDDI*, compliant with *UDDI 3.0*. The services used to test the catalogue were implemented using the *Apache CXF* framework. Only *web services*, *WSDL* and *SOAP* were supported.

Five servers were used to simulate the scenario of largely distributed repositories and CNO members. A set of 50 services was implemented in a very thin way, emulating the different activities of the several UBL processes. One hundred instances of these services were automatically generated, only and randomly varying the 11 possible QoS attributes so as to simulate the natural different services' "quality" of the CNO's providers. This total of 5000 services was equally deployed in 5 servers and also randomly registered in 5 repositories, deployed in a local network.

BPs that had some human intervention in their execution were implemented in way to provide a simple graphical interface for users to type what was required. This was inspired in the *BPEL4People* standard [12].

3.4 Catalogue Evaluation

A set of formal unit and integration tests were performed to verify the correctness of the catalogue against its requirements, especially the ones listed in Table 1.

After a sort of experiments based on many samples of UBL-based BPs modeled in the BPM environment, the systems run smoothly, supporting all the planned functionalities. In more particular, it allowed the generation and execution of new SOA/services-based applications (or changes in previously stored ones) as a result of compositions of (reused) services coming from the diverse CNO members. The intense use of open standards at all the involved levels has strongly mitigated interoperability problems and hence has facilitated the whole implementation. In other words, it was technically feasible to support compositions via the sharing of assets from the CNO members.

Considering that the developed prototype is a proof of concepts instrument handling a relatively advanced scenario, it was not feasible to test the catalogue close to a real CNO of services providers. Therefore, a more qualitative analysis had also to be done, using the *expert panel* technique [25] for that.

Nine experts on BPM and SOA were carefully selected via their curricula and previous experience on these areas for a general evaluation of the catalogue. Six experts were from the academia and the other three from IT companies, being two private and one public ones. After explaining and presenting the catalogue, they answered a questionnaire with seven questions, adopting the *Likert* scale (from *totally agree* to *totally disagree*). Some questions had a number of sub questions.

In summary, all of them agreed that: a catalogue like this can mitigate business and IT alignment problems; the catalogue is reasonably easy and intuitive to use in all of its main actions, which is suitable for SME managers; the catalogue isolates many technical details from the users when composing and generating applications; the catalogue can help companies to generate new applications in a faster and lower cost way thanks to the intense reuse. On the other hand, the interviewees expressed some concerns. In general, they were mostly related to the organizational and cultural obstacles to adopt a solution like that, both at SMEs and CNO levels. Actually, in essence, most of the identified obstacles are essentially the same than the ones pointed out in the deployment and operation of a general CNO, as depicted in [26].

4 Conclusions

This paper has presented a digital business process and software services catalogue environment as a contribution and approach to boost collaboration between CNO members of IT providers when the development of SOA/services-based applications.

Based on the research and experiments that have been carried out it was concluded that: i) an open digital catalogue environment has the potential to work as a “unified” collaborative platform for creating services-based applications within a CNO; and ii) it is technologically feasible to be built using open standards. It was also realized that the ultimate goal of a composition via the catalogue is not necessarily the development of an application for a very final concrete customer. The catalogue can also be used as a basis for: i) idealizing future applications or even acting as a supporting platform for collaborative innovation in software services, as in [27]; and ii) identifying gaps inside the CNO, which in turn can demand new services developments and can trigger other collaborations and joint results’ exploitations.

The scenario created by the catalogue environment ends up representing a win-win underlying business model that tries to take advantage of the increasing pervasiveness of providers and services. For clients, this allows to flexibly find alternative services (instead of developing them) and to bind them to their BPs considering the needed functional and non-functional requirements. For providers, their software services can be more easily discovered and more intensively used, maximizing their sustainability.

Although the model was evaluated using the UBL process model, the catalogue’s architecture is open to cope with other process models. In the same line, providers can offer new or equivalent services for different process models – even simultaneously via e.g. multi-tenant services architecture – and hence for other customers.

A number of assumptions were made to evaluate this work. The main one is that CNO partners have to adopt a common business process reference model when modeling their BPs and develop related software services. On the other hand, the adoption of conventions and models for BPs and software’s interfaces by companies and their partners is a common practice since decades. IT systems have been more and more developed using open standards to reduce interoperability problems and development costs as well as to maximize software reuse and ROI. Regarding this, it is believed that providers will be increasingly interested to develop their services following reference process models.

Next main steps of this research refer to implement an ontology for helping providers to map their services’ interfaces regarding UBL’s semantics, and to develop a resilience module to support the proper execution of the generated applications.

References

1. Ukita, 2017 <<http://www.ukita.co.uk/what-is-ukita/>> Accessed in March 2017.
2. Kramer W., Jenkins, B. The Role of the Information and Communications Technology Sector. In: Expanding Economic Opportunity Series. Report N° 22, Harvard Press (2007).
3. Camarinha-Matos, L. M., Afsarmanesh, H. A comprehensive modeling framework for collaborative networked organizations. *J. Intelligent Manufacturing*, 18(5):529-542 (2007).
4. Cancian, M. H., Rabelo, R. J., Wangenheim, C.G. Collaborative business processes for enhancing partnerships among software services providers. In: Enterprise Information Systems, 9(5-6), p. 634-659. Taylor & Francis (2015).

5. Afsarmanesh, H., Camarinha-Matos, L. M., Msanjila, S. S. Models, Methodologies, and Tools Supporting Establishment and Management of Second-Generation VBEs. *IEEE Transactions on Systems, Man and Cybernetics*, 41(5), p.692-710 (2011).
6. Adner, R. A Sad Lesson in Collaborative Innovation. *Harvard Business Review*, May 2012.
7. Van der Aalst, W. Business Process Management: A Comprehensive Survey. In: *ISRN Software Engineering* (2013).
8. Yan, Z., Dijkman, R., Grefen P. Business Process Model Repositories - Framework and Survey. In: *Journal Information Technology*, 54(4), p. 380-395 (2012).
9. Nurmilaakso, J. M. EDI, XML and e-business frameworks: a survey. In: *Computers in Industry*, 59 (4), p.370-379 (2008).
10. Perin-Souza, A., Rabelo, R. J. A Dynamic Services Discovery Model for better leveraging BPM and SOA integration. *Int.: International Journal of Information Systems in the Service Sector*, 7(1), p.1-21 (2015).
11. Järvinen, P. Action Research is Similar to Design Science. *Quality & Quantity*, 41(1), p.37-54, Springer (2007).
12. OASIS, 2017 <https://www.oasis-open.org/committees/_abbrev=ubl>.
13. Camarinha-Matos, L.M., Afsarmanesh, H., Koelmel, B. Collaborative Networks in Support of Service-Enhanced Product. In: *Adaptation and Value Creating Collaborative Networks*, p. 95-104. Springer (2011).
14. Obidallah, W.J., Raahemi, B., Kamali, S. Service oriented virtual organizations: A service change management perspective. In: *Proceedings 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering*, p.1-6. (2013).
15. Rabelo, R. J., Gusmeroli, S. The ECOLEAD Collaborative Business Infrastructure for Networked Organizations. *Pervasive Collab. Networks*, p.451-462. Springer (2008).
16. Pinheiro, F. R., Rabelo, R. J.. Experiments on Grid Computing for VE-related Applications. *Proc. 6th IFIP Working Conference on Virtual Enterprises*, p.483-492, Springer (2005).
17. The MIT Process Handbook Project (2003). In <http://ccs.mit.edu/phbook.htm>.
18. Ma, Z., Wetzstein, B., Anicic, D., Heymans, S. Semantic business process repository. In: *Proc. on Semantic Business Process and Product Lifecycle Management*, p.92–100 (2007).
19. Vanhatalo, J. IBM BPEL Repository (2005). In: [http://domino.research.ibm.com/library/cyberdig.nsf/papers/A4037428EF9DD28D85256FA5004FF88A/\\$File/rz3582.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/A4037428EF9DD28D85256FA5004FF88A/$File/rz3582.pdf)
20. Song, M., Miller, J. A., Arpinar, I. B. REPOX: an XML repository for workflow designs and specification (2001). In: <https://pdfs.semanticscholar.org/c247/13c44e48d3536130b5394415fb02274bfd83.pdf>.
21. The Oryx Project (2017). In: <http://bpt.hpi.uni-potsdam.de/Oryx>.
22. La Rosa, M., Reijers, H.A., Van der Aalst, W., APROMORE: An advanced process model repository. *Expert Systems Applications*, Vol 38, p.7029-7040 (2011).
23. Fiammante, M. *Dynamic SOA and BPM: Best Practices for Business Process Management and SOA Agility*. IBM Press (2010).
24. Rabelo, R. J., Noran, O., Bemus, P. Towards the Next Generation Service Oriented Enterprise Architecture. In: *Proceedings IEEE 19th EDOC*, p.91-100 (2015).
25. Zelkowitz, M.V., Wallace, D. R. Experimental models for validating technology. *Computer*, 31(5), p. 23-31 (2002).
26. Romero, D.; Rabelo, R. J.; Molina, A. (Orgs): Collaborative Networks as Modern Industrial Organizations: Real Case Studies. *International Journal of Computer Integrated Manufacturing*, 26(1-2). 182 p. (2013).
27. Santanna-Filho, J., Rabelo, R. J., Pereira-Klen, A. A Flexible Collaborative Innovation Model for SOA Services Providers. *Proceedings 16th IFIP Working Conference on Virtual Enterprises*, p.366-378. Springer (2015).