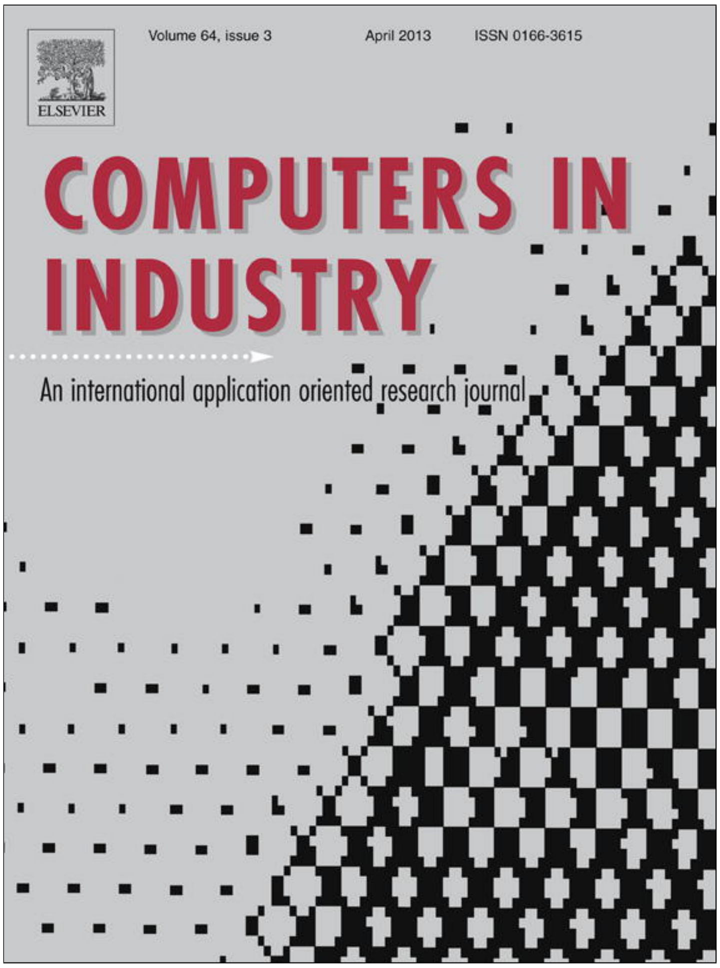

Volume 64, issue 3
April 2013
ISSN 0166-3615
ELSEVIER
COMPUTERS IN INDUSTRY
An international application oriented research journal

Computers in Industry 64 (2013) 252–267

Contents lists available at SciVerse ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A distributed repository for managing business process models in cross-organizational collaborations

Ivanna M. Lazarte [a,*], Lucinéia Heloisa Thom [c,1], Cirano Iochpe [c,2], Omar Chiotti [b,3], Pablo D. Villarreal [a,4]

[a] CIDISI, National Technological University Santa Fe Faculty, Lavaisse 610, S3004EWB Santa Fe, Argentina
[b] INGAR-CONICET, Avellaneda 3657, S3002GJC, Santa Fe, Argentina
[c] Institute of Informatics, Federal University of Rio Grande do Sul, Bento Gonçalves 91501-970, Porto Alegre, Brazil

ARTICLE INFO

Article history:
Received 22 February 2012
Received in revised form 2 November 2012
Accepted 20 November 2012
Available online 13 December 2012

Keywords:
Cross-organizational collaboration
Collaborative business process
Integration business process
Distributed business process model repository

ABSTRACT

Cross-organizational collaborations require the management of models for: collaborative business processes (CBPs), which define the collaboration's behavior; and integration business processes (IBPs), which define the behavior that supports the role an organization performs in a CBP. Managing these business process models becomes a complex task when organizations integrate collaborative networks and set up several cross-organizational collaborations. This paper presents a distributed repository that provides the functionalities required to manage conceptual business process models involved in cross-organizational collaborations. A service-oriented architecture is proposed for the distributed repository. This architecture enables organizations to access a global repository for managing collaborative networks, cross-organizational collaborations, and their CBP models. Organizations can also maintain local repositories of IBP models, which are synchronized and consistent with CBP models, while preserving their private aspects. By using verification methods and a model-driven architecture method, the distributed repository provides services that support the synchronization, consistency and interoperability requirements for CBP and IBP models. A case study is presented along with an implementation of the distributed repository.

1. Introduction

Due to globalization and the dynamism of their environment, organizations are establishing cross-organizational collaborations with their business partners in order to improve their performance and competitiveness. Cross-organizational collaborations are implemented as part of collaborative networks and supported by Information and Communication Technologies (ICT). A collaborative network consists of heterogeneous and autonomous organizations that collaborate for a period of time to achieve common goals [1–5].

The behavior of a cross-organizational collaboration is defined through *collaborative business processes* (CBPs), also called *choreographies* [6–8]. A CBP defines the interactions between organizations to achieve common business goals [9,10]. Interactions are described from a global view, i.e. through a choreography of message exchange between roles that organizations fulfill, and serves as a contractual basis for the cross-organizational collaboration [11]. CBPs are abstract processes in the sense they are not directly executable [12].

The implementation of a CBP requires each organization defines an *integration business process* (IBP), also called *private* [7], *orchestration* [6,8], or *executable* [13] process. An IBP defines the public and private activities the organization has to perform; whence, its process flow is defined from the viewpoint of one organization. Public activities (visible to external participants) support message exchange with other organizations. Private activities (hidden to external participants) comprise data transformations and/or internal functions for producing and processing the exchanged information, and they are executed by either organization's internal systems or human beings. Thus, the execution of a CBP is performed in a decentralized way by the parallel execution of the IBPs of the participants.

Conceptual IBP models are refined in several iterations to obtain IBP models with enough implementation details. Then, IBP models can be used to generate executable specifications of IBPs and interfaces of the systems of the organizations, which are required to implement (by using a process execution language,

* Corresponding author. Tel.: +54 342 4602390/4601579x258/117; fax: +54 342 4601579x215.
E-mail addresses: ilazarte@frsf.utn.edu.ar, ivanna.lazarte@gmail.com (I.M. Lazarte), lucineia@inf.ufrgs.br (L.H. Thom), ciochpe@inf.ufrgs.br (C. Iochpe), chiotti@santafe-conicet.gov.ar (O. Chiotti), pvillarr@frsf.utn.edu.ar (P.D. Villarreal).
[1] Tel.: +55 51 3308 6155.
[2] Tel.: +55 51 3308 6820.
[3] Tel.: +54 342 4534451/4535568x3007.
[4] Tel.: +54 342 4601579/4602390x258/110.

0166-3615/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.compind.2012.11.001

such as WS-BPEL [14]) and execute successfully the IBPs (by process-aware information systems [15]) according to the behavior agreed on a given CBP model. To achieve the successful execution of such IBPs, they must be consistent with its corresponding CBP and interoperable with the IBPs of other organizations participating in the CBP. *Consistency* refers to the coherence between the behavior defined in an IBP and its corresponding CBP, i.e. the logic defined in the IBP fits to the logic defined in the CBP [16]. *Interoperability*, also called *compatibility* [6,11], refers to the capability of IBPs of interacting with each other according to the behavior defined in a given CBP.

CBPs and IBPs are modeled at a conceptual level and in a technology-independent way to facilitate their design and communication among stakeholders as well as to enable their implementation in different platforms. CBP models are shared by stakeholders of different organizations, while IBP models are edited by stakeholders of an organization and maintained private into the organization's scope. *Privacy* refers to the need of avoiding organizations can access the internal business logic of other organizations. Also, both CBP and IBP models must be synchronized. *Synchronization* refers that changes realized to CBP models must be reflected into the corresponding IBP models.

Due to the need of satisfying these requirements (distributed locations, privacy, synchronization, consistency and interoperability), the management of CBP and IBP models becomes a complex and challenging task, particularly when organizations integrate several collaborative networks and set up several cross-organizational collaborations. The repository technology [17] can be applied to manage these process models. However, as discussed in Section 2, current business process model repositories [4,8,18–21] do not provide functionalities to support all the aforementioned requirements for managing process models involved in cross-organizational collaborations.

Repository system for CBP and IBP models are an important aid to contribute to a more efficient cross-organizational collaboration, helping the organizations to continually improve CBPs, and easing the change management of these processes.

To this aim, this work proposes a distributed business process model repository (distributed repository for short) for managing conceptual business process models involved in cross-organizational collaborations. The distributed repository enables the organizations a more efficient management of process models in cross-organizational collaborations, and helps them establish such collaborations more quickly and effectively by facilitating the design of IBP models, decreasing costs and development time.

The distributed repository consists of a public *global repository* and private *local repositories*. The global repository enables organizations to manage collaborative networks, cross-organizational collaborations, and their CBP models. The global repository provides the organizations with a common place to share and maintain CBP models. This repository can be provided by an organization or a neutral third part. A local repository allows an organization to manage and maintain its IBP models in a private way in order to preserve the private aspects of the organization.

To support the synchronization, consistency and interoperability requirements for CBP and IBP models, the distributed repository provides services that make use of verification methods [22,23] and an MDA-based method [24]. To prove the feasibility and utility, and to validate the repository, a real case study is presented along with an implementation of the distributed repository.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents the proposed distributed business process model repository. Section 4 presents a real case study along with an implementation of the distributed repository. Finally, Section 5 presents the contributions, the implications, the limitations, and by giving an outlook on future research.

2. Related work

A current challenge for organizations is to manage large collections of business process models. The repository technology provides the infrastructure for managing such collections [17]. Repositories providing specific functionalities for managing business process models are called *business process model repositories* [25].

Some repositories have been proposed for managing models of business processes within an organization. Advanced Process Model Repository (APROMORE) [18] offers a set of functionalities to maintain, analyze, and reuse large set of process models. The Semantic Business Process Repository (SBPR) [19] stores instances of process models based on process ontologies, and supports querying and reasoning capabilities, versioning and check-in/check-out management. The BPEL Repository [20] is an Eclipse plug-in for storing BPEL business processes and other related XML data, which provides functionalities to store, find and use these documents.

There are other proposals that enable organizations managing business process models in cross-organizational collaborations. The Collaborative Interoperability Framework (CIbFw) [4] supports seamless interoperability in a networked environment by enabling organizations, which have submitted their collaboration profile to a centralized repository, to be added to and removed from a collaborative network and to perform e-business by exchanging business documents. CIbFw is based on a service-oriented approach. The ebXML Registry [8] enables organizations to register global and local choreographies (public processes) maintaining the dependencies between these choreographies. The BPMN Repository Architecture [21] enables planning, implementing and controlling cross-organizational processes. The core component of this architecture is a distributed repository managing all required data and information.

Table 1 summarizes results of comparing these repositories with regards to their ability to meet the requirements (discussed in Section 1) for managing business process models in cross-organizational collaborations. These requirements are: distributed locations, privacy, synchronization, consistency and interoperability. Support for process model design was also considered.

APROMORE, SBPR, and BPEL Repository are focused on managing business process models that do no exceed the boundaries of an organization. They are not implemented in distributed environments.

APROMORE supports the consolidation of process models [26] through an approach to semi-automatically aggregate a collection of process models into a single one (called a *merged model*). Changes to the merged model are propagated to each affected variant of a business process, keeping them synchronized. However, it does not consider the synchronization between different types of business process models, such as CBP and IBP models. SBPR and BPEL Repository do not provide support for such

Table 1
Comparison of business process model repositories.

Comparison criteria	Business process model repositories					
	[18]	[19]	[20]	[4]	[8]	[21]
Cross-organizational domain	–	–	–	+	+	+
Distributed location	–	–	–	–	–	+
Synchronization	+–	–	–	–	–	–
Interoperability	–	–	–	–	+–	–
Consistency	+–	–	–	–	+–	–
BP model design support	+–	+–	+–	+–	+–	+–
Privacy	+	+	+	+	+	+

+, supported; +–, partially supported; –, not supported.

synchronization. Also, these three repositories do not support the interoperability between business process models.

SBPR and BPEL Repository do not provide support for consistency checking. APROMORE identifies the conformance analysis, as a type of consistency checking, to evaluate to which extent an input model conforms to a reference process model in a given domain. However, this analysis is not oriented to be applied to check consistency between different types of process models, such as CBP and IBP models.

Support for business process model design is provided by these three repositories based on reusing existing content, i.e. reuse of business process fragments or whole business processes. This is appropriate for bottom-up development approaches of cross-organizational collaborations, in which each organization first defines its IBPs in isolation, and then it must search for potential business partners whose processes are complementary to each other. However, discovering potential business partners requires complex comparisons of IBPs and is rather unlikely to find complementary IBPs [8]. This work focuses on cross-organizational collaborations that start from collaborative agreements and common business goals.

APROMORE, SBPR, and BPEL Repository preserve the organizations' private aspects due to they maintain process models within an organization.

CIbFw, ebXML Registry, and BPMN Repository Architecture are focused on managing business process models in cross-organizational collaborations. But only the last of them has a distributed implementation.

Synchronization between business process models is not supported by any of these three repositories. Although ebXML Registry maintains the dependencies between global and local choreographies, it does not maintain their synchronization.

ebXML Registry guarantees consistency and interoperability between global and local choreographies due to each organization derives its local choreography from a common global choreography based on a dedicated UML profile. A local choreography describes a public process from the viewpoint of an organization without indicating its private activities. Therefore, IBP models are not generated and there is not a provided mechanism to check consistency and interoperability for IBP (or orchestration) models. CIbFw and BPMN Repository Architecture do not provide the support for consistency and interoperability between business process models.

CIbFw enables organizations to reuse and download collaboration agreement templates, fill them in, and upload them to a centralized repository. ebXML Registry provides design support by enabling the reuse of available global choreographies. BPMN Repository Architecture manages reference models that can be imported from the repository into modeling tools to construct individual process models. However, these approaches do not provide mechanisms to generate and design interoperable and consistent IBP models from CBP models.

CIbFw and BPMN Repository Architecture preserve the organizations' private aspects through access control on the repository's data. ebXML Registry achieves it by storing only global and local choreographies, which do not include private business logic of the organizations.

In summary, the aforementioned business process model repositories do not provide functionalities to support all requirements for managing process models involved in cross-organizational collaborations, in particular the requirements of synchronization, interoperability and consistency.

3. A distributed business process model repository

This section presents a distributed repository for managing conceptual business process models defined at a design time for cross-organizational collaborations. The distributed repository consists of: a public *global repository* for managing CBP models, and private *local repositories* for managing IBP models. The aim is twofold: (1) to allow organizations sharing and managing CBP models they have designed to define the behavior of cross-organizational collaborations, by using the public global repository; (2) to enable organizations having private local repositories, synchronized with the global repository, for maintaining IBP models along with details about requirements the organizations have to fulfill to implement the defined CBPs. This separation of related business process models in different repositories makes it possible to deal with the particular requirements for business process model repositories: distributed locations, privacy, design support, synchronization, consistency and interoperability of process models. Hence, the proposed distributed repository provides the following main functionalities:

- *Synchronization of distributed business process models*: the synchronization among a CBP model of the global repository and its IBP models maintained in the local repositories must be kept to ensure that changes on the CBP model (new versions) are propagated and translated to existing IBP models or to new generated versions of IBP models. Thus, organizations can use the right version of models to evaluate, redesign and implement CBPs and IBPs.
- *Checking of consistency between business process models*: by ensuring the business logic defined in an IBP model of an organization is consistent with the business logic defined in the corresponding CBP model. This allows organizations to define and maintain in their local repositories IBP models that are in accordance with the business logic agreed on the CBPs.
- *Guarantee of interoperability between business process models*: by determining if the public message exchange of IBP models of organizations is synchronized. This guarantees that IBPs can interact and successfully execute the corresponding CBPs.
- *Preservation of private aspects of organizations*: by enabling each organization to manage its IBP models in a private way in its local repository, maintaining these models hidden and protected from the access to other organizations.
- *Business process design support*: by assisting organizations in the design of CBP and IBP models. The global repository enables the management of catalogs of CBP models and provides support to search into the catalogs and select CBP models that can be reused or adapted for creating new CBP models. Also, the global repository enables the automatic generation of IBP template models from a CBP model and their transfer and storing in the local repositories of the organizations. From an IBP template, an organization can define an IBP model by maintaining the skeleton given by the template and just modifying or replacing the suggested abstract private activities by the concrete or final private activities the organization defines. Thus, the repository also provides features that enables organizations to design CBP and IBP models more easily and quickly.

The distributed repository is based on a service-oriented architecture (SOA) [27,28], with the aim of providing distributed, interoperable, reusable and loose-coupled services that support functionalities for the global and the local repositories (Fig. 1). The *global repository* is concerned with the management of collaborative networks, cross-organizational collaborations and CBP models. This can be accessed and managed by the organizations registered in the repository. A *local repository* is concerned with the management of IBP models that an organization requires to carry out the definition, management and implementation of CBP models. A local repository can be accessed and managed only by the owner organization. The aim is to preserve the autonomy of the

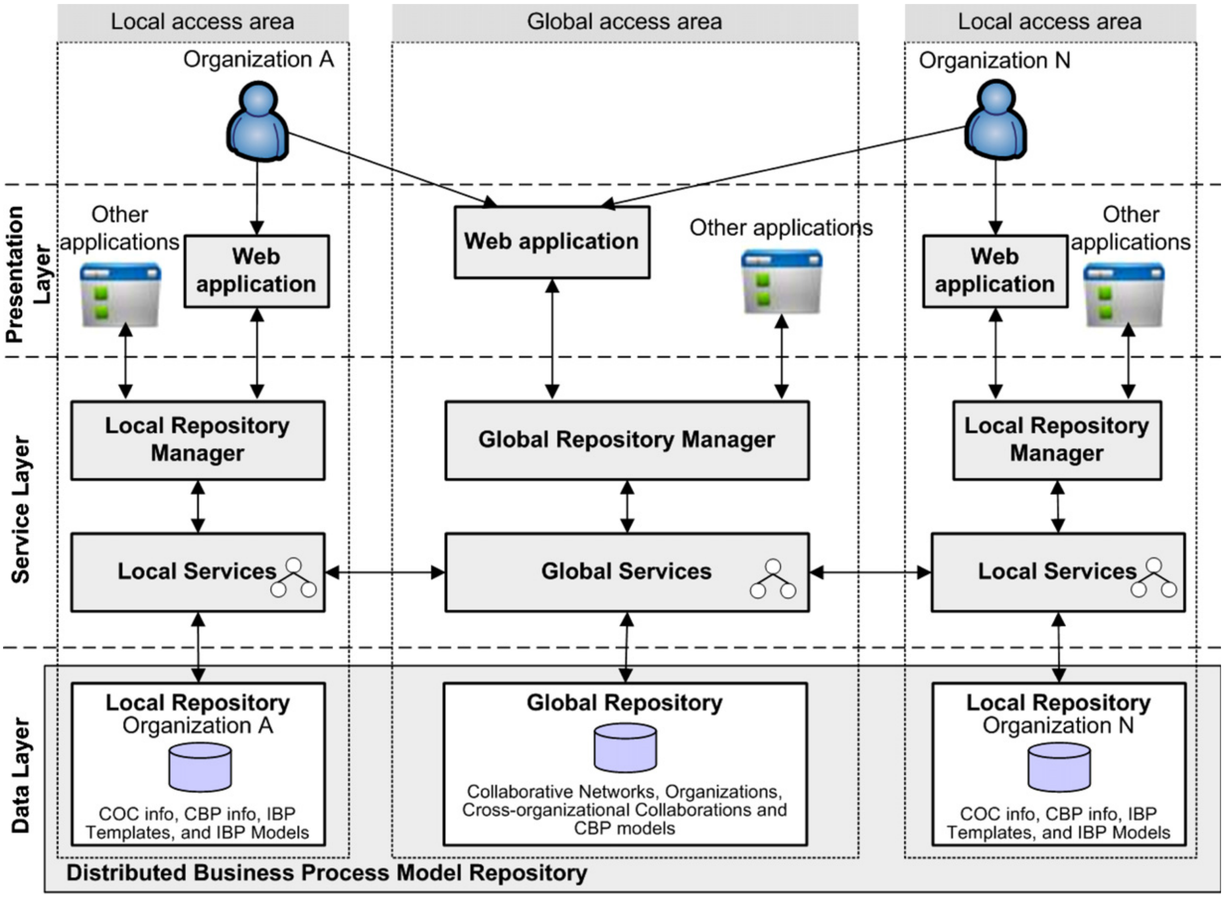


Fig. 1. Architecture of the distributed business process model repository.

organizations and their private aspects, and to enable the decentralized and distributed management of the process models involved in cross-organizational collaborations.

The architecture of the distributed repository follows a three-tier model which consists of: a *presentation layer*, a *service layer* and a *data layer*. The presentation layer consists of client applications providing user interfaces so that users can have access to the distributed repository. The service layer provides all services required for the management of business process models of the distributed repository. The data layer stores business process models and relevant data related to them. These layers are described in more detail below.

3.1. The data layer

Fig. 2(a) and (b) shows the conceptual models of the entities that are persisted in the global repository and in a local repository, respectively.

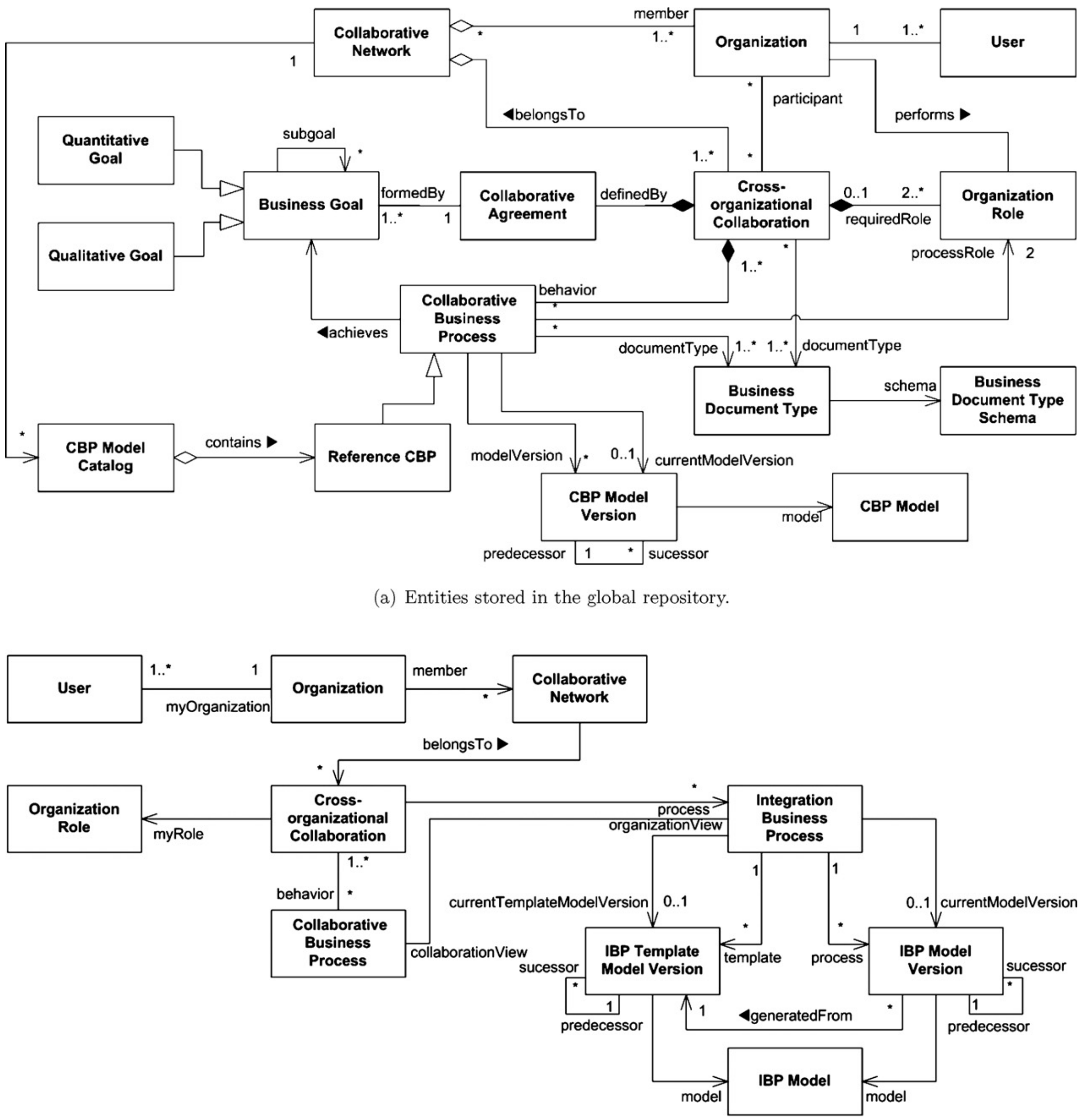
A *global repository* contains a registry of collaborative networks and the organizations that are member of them. Users of the global repository are people of these organizations.

A collaborative network is a group of organizations related to a specific domain whose aim is to define and participate in cross-organizational collaborations. A collaborative network consists of the organizations that are members and the cross-organizational collaborations these organizations have agreed to carry out. A collaborative network can have catalogs of CBP models, which contain reference CBPs with their models. A reference CBP can be obtained from reference business models that are established by industry initiatives (e.g. CPFR [29]) for particular domains, and provides explicit knowledge about how to carry out the collaboration. This knowledge can be reused and adapted to define specific CBP models.

A cross-organizational collaboration represents a set of organizations that cooperate to achieve common business goals through the joint definition, implementation, execution and evaluation of CBPs. A cross-organizational collaboration consists of the organizations that participates in it, the role each organization performs, a collaborative agreement, and the CBPs the organizations have agreed to carry out. A collaborative agreement defines parameters that govern a cross-organizational collaboration, such as the reference business model adopted and the duration of the collaboration [4,9]. It also has an associated hierarchy of common business goals the organizations must achieve. A business goal can be defined either quantitatively or qualitatively. Goals are quantitatively defined through performance measurements about some information shared by organizations. Goals are qualitatively defined through an informal description, and they depend on human judgment rather than a specific value.

Business document types can be defined in a cross-organizational collaboration and they refer to the information to be exchanged by the participant organizations in the CBPs. A business document type defines the information to be included and the overall structure of the document, and it can be used in different CBPs. Its structure can be defined by referencing the schema of a document of a B2B standard (e.g. BODs [30], UBL [31]) or ad hoc by the involved organizations. The structure of the document is defined generally in an XML document associated to the business document type and stored into the repository.

The organization roles each organization has to perform in a cross-organizational collaboration are associated to each CBP



(a) Entities stored in the global repository.

(b) Entities stored in a local repository.

Fig. 2. Entities stored in the distributed repository (association cardinalities of 1 are omitted). (a) Entities stored in the global repository and (b) entities stored in a local repository.

stored into the repository. In this way, through the cross-organizational collaborations, organizations involved in the CBPs and their roles can be known. The business goal to be achieved by a CBP can also be associated to it according to the business goals defined in the cross-organizational collaboration. It is also possible to define business document types used in the CBPs, which must correspond with those defined in the cross-organizational collaboration. The CBP entity can store as attribute an informal description of the process, which includes the steps of both a main scenario and alternative scenarios. The behavior of a CBP is explicitly defined in its model versions.

The versioning allows maintaining the evolution of a CBP model in the global repository. Several versions of a CBP model can be defined. Just one version can be marked as *current model version* of a CBP. The repository allows defining a hierarchy of model versions, which enables representing tracks of changes on the model versions, and for each model version, indicate its source model version (predecessor) and the model versions (successors) derived from it. A model version describes details such as creation date, version number, description and so on, and contains the CBP model it represents.

A *local repository* of an organization contains replicated information of the global repository about collaborative networks and cross-organizational collaborations in which the organization participates, the organization role performed in each cross-organizational collaboration, and the CBPs that the organization is involved in and supports.

For each CBP, an integration business process (IBP) is maintained, which specifies both the public and private behavior of the organization, required to implement and support the CBP. An IBP template model represents a blueprint of an IBP, which is automatically generated by the distributed repository, and serves as a skeleton to define an IBP model. An IBP template model has both public activities derived from a CBP model and abstract private activities. An IBP model has the complete behavior of the IBP and it is derived from an IBP template model by redefining the abstract private activities through concrete private activities.

The versioning allows maintaining the evolution of IBP templates and IBP models into the local repository. The local repository allows defining a hierarchy of model versions for IBP templates and IBPs. An IBP model version has associated the IBP template model version used for generating it.

Users of a local repository are people of the organization that is owner of the repository.

3.2. The service layer

The service layer provides services that support functionalities of the distributed repository to manage business process models. This layer consists of a *global repository manager* and *local repository managers*, one for each organization (Fig. 1). The global repository manager acts as a façade providing a unified interface for client applications to manage CBP models. A local repository manager acts as a façade providing a unified interface for client applications to manage IBP models. Both repository managers interact with basic and specific services that contain the logic of functions of the distributed repository. Basic services support basic functions commonly provided by both general repositories [32] and business process repositories [25], which are:

- *Storage service*: to create, update, and delete process models.
- *Retrieval service*: to obtain the required process model according to some criteria through navigation, query, and search methods.
- *Integration service*: to enable the integration of external tools to the repository.

Table 2
Main specific services provided by the global repository.

Service	Service operation
OrganizationManagement	addOrganization(org:Organization) getOrganization(orgInfo:OrgInfo):org:Organization getOrganizations(searchCriteria:OrgSearchCriteria):organizations:OrgList removeOrganization(orgInfo:OrgInfo) updateOrganization(orgInfo:OrgInfo, org:Organization)
CNManagement	addCN(cn:CollaborativeNet) updateCN(cnInfo:CNinfo, cn:CollaborativeNet) removeCN(cn:CnInfo) getCN(cn:CnInfo):cn:CollaborativeNet addMember(cn:CnInfo, orgInfo:OrgInfo) removeMember(cnInfo:CNinfo, orgInfo:OrgInfo) getMember(cnInfo:CNinfo, orgInfo:OrgInfo): org:Organization getMembers(cnInfo:CNinfo):memberList:OrgList
COCManagement	addCOC(cnInfo:CNinfo, coc:COC) getCOC(cocInfo:COCinfo):coc:COC removeCOC(cocInfo:COCinfo) updateCOC(cocInfo:COCinfo, coc:COC) addParticipant (cocInfo:COCinfo, role:OrganizationRole, orgInfo:OrgInfo) removeParticipant (cocInfo:COCinfo, orgInfo:OrgInfo) getParticipant(cocInfo:COCinfo, role:OrganizationRole): org:Organization getParticipants(cocInfo:COCinfo): organizations:OrgList getOrganizationRole(cocInfo:COCinfo, orgInfo:OrgInfo): role: OrganizationRole getOrganizationRoles(cocInfo:COCinfo, orgInfo:OrgInfo): roles: OrganizationRoleList
CollaborativeAgreementManagement	addCollaborativeAgreement(cocInfo:COCinfo, ca: CollaborativeAgreement) removeCollaborativeAgreement(cocInfo:COCinfo, caInfo: CAinfo) updateCollaborativeAgreement(cocInfo:COCinfo, caInfo: CAinfo, ca: CollaborativeAgreement) getCollaborativeAgreement(cocInfo:COCinfo, caInfo: CAinfo):ca: CollaborativeAgreement addBusinessGoal(caInfo: CAinfo, bg: BusinessGoal) removeBusinessGoal(caInfo: CAinfo, bgInfo: BusinessGoalInfo) updateBusinessGoal(caInfo: CAinfo, bgInfo: BusinessGoalInfo, bg: BusinessGoal) getQuantitativeBusinessGoals(caInfo: CAinfo) getQualitativeBusinessGoals(caInfo: CAinfo) getBusinessGoal(caInfo: CAinfo, bgInfo: BusinessGoalInfo)
CBPCatalogManagement	addCBPCatalog(cnInfo:CNinfo,cbpCatalog: CBPCatalog) updateCBPCatalog(cnInfo:CNinfo, cbpCatalogInfo: CBPCatalogInfo, cbpCatalog: CBPCatalog) removeCBPCatalog(cnInfo:CNinfo, cbpCatalogInfo: CBPCatalogInfo) getCBPCatalog(cbpCatalogInfo: CBPCatalogInfo): cbpCatalog:CBPCatalog getCBPCatalogs(cbpCatalogInfo: CBPCatalogInfo): cbpCatalogList:CBPCatalogList addReferenceCBP (cbpCatalogInfo: CBPCatalogInfo, referenceCBP:ReferenceCBP) getReferenceCBP(cbpID: ReferenceCBPInfo): referenceCBP: ReferenceCBP getReferenceCBPmodels(cbpCatalogInfo: CBPCatalogInfo, searchCriteria:CBPSearchCriteria):referenceCBPList:CBPList removeReferenceCBP(cbpCatalogInfo: CBPCatalogInfo, referenceCBPinfo:CBPinfo) updateReferenceCBP(cbpCatalogInfo: CBPCatalogInfo, referenceCBPinfo:CBPinfo, referenceCBP:ReferenceCBP)
CBPManagement	addCBP(cocInfo:COCinfo, cbp: CollaborativeBusinessProcess) removeCBP(cocInfo:COCinfo, cbpInfo: CBPinfo) updateCBP(cocInfo:COCinfo, cbpInfo: CBPinfo, cbp: CollaborativeBusinessProcess) getCBPs(cocInfo:COCinfo):cbpList:CBPList getCBP(cbpInfo: CBPinfo): cbp: CollaborativeBusinessProcess getOrganizationRoles(cbpInfo: CBPinfo): roles: OrganizationRoleList

- *Access management service*: to ensure that users only have access to the repository that they are authorized to access.
- *Checkin/checkout service*: to allow a user to check-out objects from the repository, lock them so others cannot change them, make changes and check them in again by releasing the lock.
- *Notification service*: to generate notifications in case a process model in the repository is changed.
- *Version management service*: to enable multiple versions of a process model.
- *Configuration management service*: to maintain the relation between (a version of) a process model and the (version of) subprocess models that it consists of.
- *Lifecycle management service*: to maintain the stage (deprecated, validation, current) of the lifecycle that a process model is currently in.

Specific services support the particular functions aforementioned required to manage business process models in cross-organizational collaborations, which are described in the next subsections.

Tables 2 and 3 show an overview of the main specific services provided by the global and local repository, respectively, indicating their service operations.

3.2.1. Services of the global repository

This section describes the specific services provided by the global repository (Table 2).

3.2.1.1. OrganizationManagement service. The `OrganizationManagement` service is responsible for managing the organizations registered in the repository. This service provides operations to add/update/remove/get organizations involved in the repository. Before organizations participate in a collaborative network, firstly they must load their profile via the Web application which derives in the invocation of the add method of this service.

3.2.1.2. CNManagement service. The `CNManagement` service is responsible for managing collaborative networks. This service provides operations to add/update/remove/get collaborative networks, add/remove/get organizations that are members of a collaborative network, and invite organizations to join a collaborative network.

3.2.1.3. COCManagement service. The `COCManagement` service is responsible for managing cross-organizational collaborations. This service provides operations to add/update/remove/get cross-organizational collaborations. Also, it provides operations to add/update/get organizations that participate in a cross-organizational collaboration and the roles they fulfill.

3.2.1.4. CollaborativeAgreementManagement service. The `CollaborativeAgreementManagement` service is responsible for managing the collaborative agreement defined for a cross-organizational collaboration. This service provides operations to add/update/remove/get collaborative agreements, and business goals associated to a collaborative agreement.

3.2.1.5. BusinessDocumentTypeManagement service. The `BusinessDocumentTypeManagement` service is responsible for managing business document types to be exchanged in each CBP model. This service provides operations to add/update/remove/get business document types and their schema.

3.2.1.6. CBPCatalogManagement service. The `CBPCatalogManagement` service is responsible for managing catalogs of reference CBPs of a cross-organizational collaboration, which generally correspond to a reference business model. This service also provides operations to add/update/remove/get reference CBPs of a catalog.

3.2.1.7. CBPManagement and CBPModelManagement services. The `CBPManagement` service provides operations to add/update/remove/get CBPs of a cross-organizational collaboration.

The `CBPModelManagement` service is responsible for managing model versions of CBPs. This service has operations to get model versions of a CBP, add a model version and set the current CBP model version.

Table 3
Main specific services provided by a local repository.

Service	Service operation
IBPManagement	addIBP(cocInfo:COCinfo,cbpInfo: CBPinfo, ibp:IntegrationBusinessProcess) removeIBP(cocInfo:COCinfo, ibpInfo:IBPinfo) getIBPs(cocInfo:COCinfo)ibpList:IBPList getIBP(ibpInfo:IBPinfo): ibp:IntegrationBusinessProcess
IBPTemplateModelVersionManagement	addIBPTemplateModelVersion(ibpInfo:IBPinfo, ibpTemplateModel:IBPTemplateModel, predecessorIBPTemplatModelVersion: IBPTemplateModelVersion: newIBPTemplateModelVersion: IBPTemplateModelVersion) setCurrentIBPTemplateModelVersion(ibpInfo: IBPinfo, ibpTemplateModelVersion: IBPTemplateModelVersion) getCurrentIBPTemplateModelVersion(ibpInfo: IBPinfo): ibpTemplateModelVersion: IBPTemplateModelVersion getIBPTemplateModelVersions(ibpInfo: IBPinfo): ibpTemplateModelVersionList: IBPTemplateModelVersionList getPredecessorIBPTemplateModelVersion(ibpTemplateModelVersion: IBPTemplateModelVersion): predecessorIBPTemplateModelVersion: IBPTemplateModelVersion getSuccessorIBPTemplateModelVersions(ibpTemplateModelVersion: IBPTemplateModelVersion): sucessorIBPTemplateModelVersions: IBPTemplateModelVersionList
IBPModelVersionManagement	addIBPModelVersion(ibpInfo:IBPinfo, ibpModel:IBPmodel, predecessorIBPModelVersion: IBPModelVersion: newIBPModelVersion: IBPModelVersion) setCurrentIBPModelVersion(ibpInfo: IBPinfo, ibpModelVersion: IBPModelVersion) getCurrentIBPModelVersion(ibpInfo: IBPinfo): ibpModelVersion: IBPModelVersion getIBPModelVersions(ibpInfo: IBPinfo): ibpModelVersionList: IBPModelVersionList getPredecessorIBPModelVersion(ibpModelVersion: IBPModelVersion):predecessorIBPModelVersion: IBPModelVersion getSuccessorIBPModelVersions(ibpModelVersion: IBPModelVersion): sucessorIBPModelVersions: IBPModelVersionList
IBPmodelConsistencyChecking	verifyIBPModelConsistency(ibpTemplateModel:IBPmodel, ibpModel:IBPmodel): verificationResult:VerificationResult
LocalSynchronization	storeCOCinfo(cocInfo:COCinfo) storeCBPinfoAndIBPTemplate(cocInfo:COCinfo, cbpInfo:CBPinfo, ibpTemplateModel:IBPmodel) deprecateIBPtemplate(cocInfo:COCinfo, cbpInfo:CBPinfo)

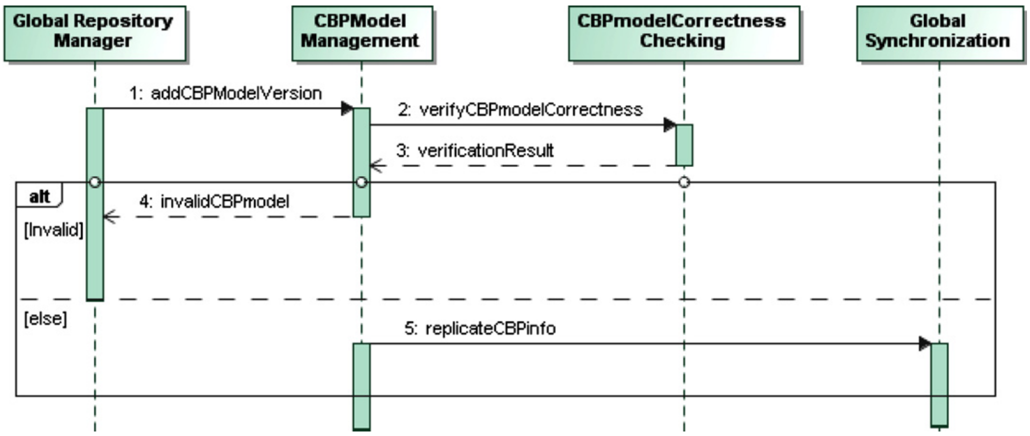


Fig. 3. Interaction between services required to add a CBP model (input/output of each operation is omitted).

The addCBPModelVersion operation enables the addition of a new model version to a CBP, by indicating the predecessor CBP model version from which the new model version will be put in the hierarchy of model versions. This operation invokes other services, as it is illustrated in Fig. 3. Firstly, an operation of the CBPmodelCorrectnessChecking service is invoked to verify and determine if the behavior defined in the CBP model is error-free, previous to the generation of the IBPs of the organizations. If the verification result is not ok, the addition of the CBP model version is canceled and the addCBPModelVersion operation returns an invalidCBPmodel fault message, indicating errors found in the CBP model. Else, if the verification result is ok, the replicateCBPinfo operation of the GlobalSynchronization service is invoked in order to carry out the synchronization of the CBP with the IBP template models of the corresponding local repositories of the organizations involved in the CBP.

3.2.1.8. CBPmodelCorrectnessChecking service. The CBPmodelCorrectnessChecking service is responsible for ensuring the behavior of the CBP is well-defined. This service provides the verifyCBPmodelCorrectness operation that verifies the correctness of a given CBP model, such as the absence of deadlocks and livelocks.

To do this, this operation implements the CBP model verification method proposed by Roa et al. [22]. This method formalizes the global view of interactions of CBP models by means of Global Interaction Nets (GI-Net), which is a particular type of Hierarchical and Colored Petri Net. The method uses a Soundness property for GI-Nets as the main correctness criterion to verify advanced control flows in CBP models. An important feature of this verification method is that it is independent of the modeling language whose models are going to be verified, i.e. it can be applied to verify models defined with any CBP modeling language.

3.2.1.9. IBPTemplateGeneration service. The IBPTemplateGeneration service is responsible for facilitating the design of IBP models. This service provides the generateIBPTemplate operation for generating an IBP template model from a CBP model. This template represents the skeleton of the expected behavior of an organization required to perform a role it fulfills in the CBP model. Then, organizations refine an IBP template model to obtain an IBP model.

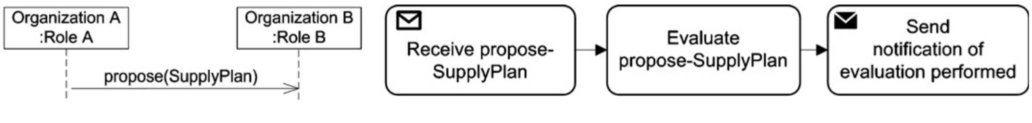
For the automatic generation of IBP template models, this service implements a method based on the Model-Driven Architecture (MDA) [33] that was proposed in previous work [24]. This method provides a model-to-model transformation process that takes as inputs a CBP model and a target organization role, and automatically generates as output an IBP template model that contains the public and private activities and control flow logic that an organization has to implement for performing the target role. The input CBP model is based on the UP-ColBPIP language [9,34], and the output model is based on the BPMN language [7]. These languages allow representing platform-independent process models at a conceptual level. By using the UP-ColBPIP language the behavior of CBPs is modeled through interaction protocols. An interaction protocol describes a choreography of business messages between organizations that play different roles. Thus, it is a message-oriented structured language that also provides semantics to the cross-organizational messages via the use of speech acts, which indicate the intention of the message's sender with regard to a business document being exchanged. More details about this language can be found in [9,34].

An overview of the main steps of the method to transform a CBP model (represented as an interaction protocol) into an IBP template model is given in Algorithm 1. The transformation process consists on analyzing each element of an interaction protocol M_{CBP} from the viewpoint of a selected role R of the protocol, and generating public/private logic and activities in the output IBP template model M_{IBP} by applying transformation rules TR. The input of a rule is a protocol element $e \in M_{CBP}$. The output of a rule is a predefined BPMN pattern $p \in P$ that expresses the semantics of a protocol element in terms of elements and semantics provided by BPMN language.

Algorithm 1. Algorithm to generate an IBP template M_{IBP} for a given CBP model M_{CBP} and a role R.

Input: M_{CBP} and R
  create a BPMN diagram d
  for each element $e \in M_{CBP}$ and viewpoint R **do**
    transform element e into pattern $p \in P$
    insert pattern p into diagram d
  end for
  $M_{IBP} \Leftarrow d$
Output M_{IBP}

The output patterns of the rules that transform business messages of a protocol were defined according to workflow activity patterns [35], which represent recurrent business functions frequently found in business processes. The semantics of the business messages is essential to identify the workflow activity



(a) LHS of the rule. (b) RHS of the rule.

Fig. 4. Graphical representation of the rule for transforming a business message whose intention is *propose*. (a) LHS of the rule and (b) RHS of the rule.

pattern to be used in the transformation [24]. The use of workflow activity patterns ensures the interoperability in the message exchange between IBPs by providing a synchronization among the receiving activities and the sending activities generated in the IBPs.

For example, an organization B (fulfilling the role B) receives a business message whose intention is to propose a SupplyPlan that is exchanged through the business document attached to the message (Fig. 4(a)). This message is transformed into three tasks (Fig. 4(b)) according to the Bi-directional Performative workflow pattern [35]: a public *Receive Task* [7] labeled as *Receive propose-SupplyPlan*, an *Abstract Task* [7] labeled as *evaluate propose-SupplyPlan* that represents the private logic to be carried out in the organization, and a public *Send Task* [7] labeled as *Send notification of evaluation performed*. The business document is mapped into a *Message* labeled as *propose-SupplyPlan*, associated to the Receive Task.

A generated IBP template model contains: (a) public activities (Send and Receive Tasks [7]) that enable an organization to send or receive business messages defined in the CBP model, (b) abstract private activities (Abstract Tasks [7]) that enable the organization to process or generate the information to be exchanged, and (c) the control flow (Gateways [7]) that enables the organization to carry out the routing of business messages as it was defined in the CBP model. The generated public activities and the control flow related to these activities must remain inalterable to preserve the consistency between the CBP model and the IBP template and guarantee interoperability between IBP models of each organization. Private activities are generated as abstract, and therefore they must be refined or decomposed in concrete private subprocesses or activities when an IBP model is defined from the IBP template model. In the previous example (Fig. 4(b)), business analysts must refine the *evaluate propose-SupplyPlan* abstract task in order to indicate how the organization carries out the evaluation.

Thus, this method guarantees the generated IBP template model is consistent with the input CBP model, and hence interoperable with the IBP template models of the other organizations involved in the CBP (Fig. 6(a)). More details about this MDA-based method for generating IBP template models can be found in [24].

3.2.1.10. GlobalSynchronization service. The `GlobalSynchronization` service is responsible for replicating the information from the global repository to a local repository. This service provides the `replicateCOCinfo`, `replicateCBPinfo`, and `notifyCBPdeprecated` operations.

The `replicateCOCinfo` operation replicates in a local repository the information about a cross-organizational collaboration when an organization is added as participant of it into the global repository.

The `replicateCBPinfo` operation replicates in the local repository of an organization the information about a CBP and the respective IBP template model, when it is added in a cross-organizational collaboration where the organization is involved. The logic of this operation and the interaction between services is illustrated in Fig. 5. Firstly, the `CBPManagement` service is invoked to get the organization roles involved in the CBP added. Then, for each role, the `COCManagement` service is invoked to get the organization performing the role in order to send it the corresponding information. Following, the `IBPTemplateGeneration` service is invoked to generate the IBP template model, and finally, the `LocalSynchronization` service of the participant organization is invoked to store the information about the CBP and the IBP template into the local repository.

The `notifyCBPdeprecated` operation is responsible for notifying to the `LocalSynchronization` service of local repository of each organization that a CBP model became deprecated, so that they set to `deprecated` the status of the respective IBP template models and, therefore, they cannot be used for generating IBP models.

3.2.2. Services of a local repository

This section describes specific services provided by each local repository (Table 3).

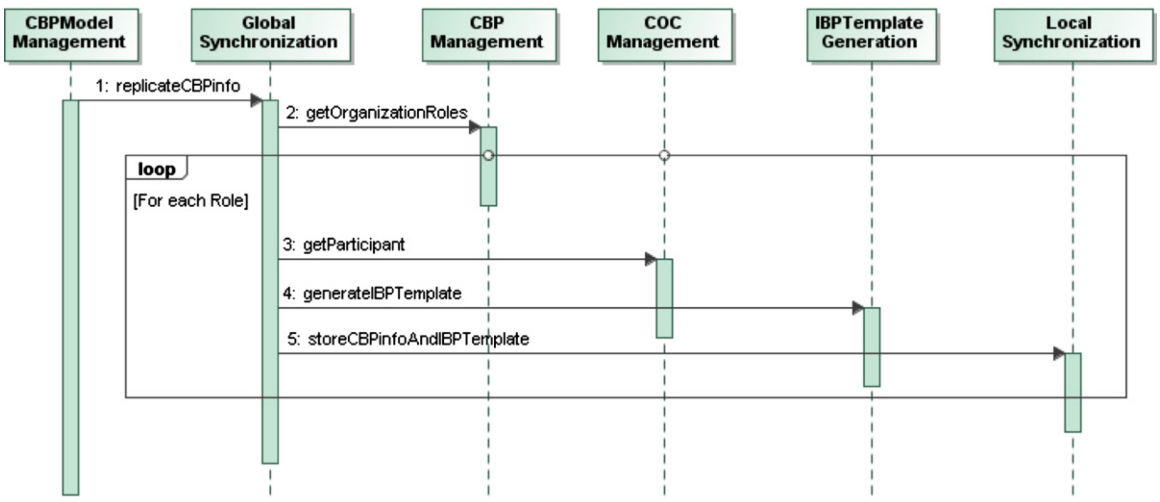


Fig. 5. Interaction between services required to synchronize the models when a CBP model is added (input/output of each operation is omitted).

3.2.2.1. IBPManagement service. The `IBPManagement` service provides operations to add/remove/get IBPs in the local repository.

3.2.2.2. IBPTemplateModelVersionManagement service. The `IBPTemplateModelVersionManagement` service is responsible to manage model versions of IBP templates. This service provides operations to add and get model versions of an IBP template, and set the current IBP model version.

3.2.2.3. IBPmodelVersionManagement service. The `IBPmodelVersionManagement` service is responsible for managing model versions of IBPs. This service provides operations to add and get model versions of an IBP, and set the current IBP model version.

3.2.2.4. IBPmodelConsistencyChecking service. As it was introduced in Section 3.1, each organization involved in a CBP must define its IBP model by redefining the IBP template model. This means taking the structure of the IBP template model and just redefining its abstract private activities through concrete private activities, which represent how really the organization will carry out its private business requirements. This is an error prone task that can generate incompatible IBP models, i.e. IBPs that cannot interoperate.

Therefore, a consistency checking is required in order to guarantee the interoperability of IBP models of the participant organizations, and consequently, the correct termination of a CBP. Due to interoperability and consistency are related notions [6,11], by checking if an IBP model is consistent with its IBP template model generated from a CBP model (Fig. 6(b)), the interoperability between IBP models of different organizations is insured, i.e. whenever they in turn are consistent with their respective IBP template. In this way, each organization can locally check if an IBP model is consistent with the CBP model agreed on a cross-organizational collaboration. The consistency checking between IBP template models and its corresponding CBP model is not required due to IBP templates are automatically generated from CBP models by applying the model-to-model transformation process described in Section 3.2.1.9 [24].

The `IBPmodelConsistencyChecking` service supports the verification of the consistency between an IBP model with its IBP template model, by comparing their behaviors. This service provides the `verifyIBPModelConsistency` operation, which implements the behavior consistency verification method proposed by Martens in [23]. Although this method was proposed to get consistency between two different types of business process, i.e. between a BPEL-based abstract process [14] and a BPEL-based executable process [14], its verification based on the comparison of behavior of two process models is here reused. This distributed repository's service reuses the logic of this method to compare two IBP models, one that is a template against with the another one whose skeleton is the template.

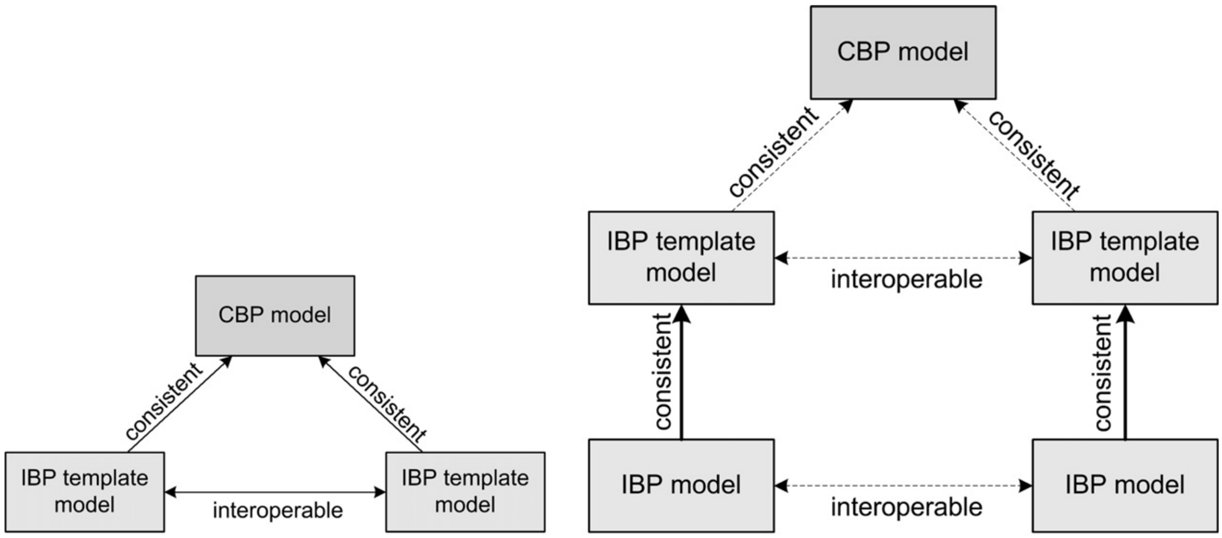
Therefore, by applying this method, an IBP model is called to be consistent with an IBP template model if and only if the behavior of the former simulates the behavior of the latter.

This verification method analyzes the consistency between an IBP template model and an IBP model derived of the template in three steps [23]:

1. Transforming both IBP models into a Petri net. Considering that we use BPMN for modeling IBPs instead of WS-BPEL, we adjust this step of the original method to accept process models defined with the BPMN language. The mapping of a BPMN process model to Petri nets is realized by applying the mapping rules proposed by Dijkman et al. [36].
2. Extracting the relevant information and generating the *communication graph* [23,37] for each IBP model.
3. Verifying the simulation of behavior by comparing the corresponding communication graphs.

An IBP model simulates an IBP template model if an only if the communication graph of the IBP model simulates the communication graph of the IBP template model.

3.2.2.5. LocalSynchronization service. The `LocalSynchronization` service keeps the synchronization between IBP template models and CBP models stored in the global repository. The aim is



(a) Consistency and interoperability relationships between models produced by the `IBPTemplateGeneration` service.

(b) Consistency relationships to be checked by the `IBPmodelConsistencyChecking` service.

Fig. 6. Consistency and interoperability relationships between process models involved in a cross-organizational collaboration. (a) Consistency and interoperability relationships between models produced by the `IBPTemplateGeneration` service and (b) consistency relationships to be checked by the `IBPmodelConsistencyChecking` service.

to ensure that the version of IBP models used by organizations corresponds to the right version (latest or previous) of the respective CBP model. This service provides the `storeCOCinfo`, `storeCBPinfoAndIBPTemplate`, and `deprecateIBPtemplate` operations.

The `storeCOCinfo` operation is responsible for receiving information about the cross-organizational collaboration in which the organizations is being added as participant in order to store such information into the local repository.

The `storeCBPinfoAndIBPTemplate` operation is responsible for receiving the IBP template model along with information about the corresponding CBP, storing the CBP information, and invoking the `IBPManagement` and `IBPTemplateModelVersionManagement` service to add the IBP template model into the local repository.

The `deprecateIBPtemplate` operation is responsible for receiving information about the deprecated CBP model in order to set the `deprecated` status to the corresponding IBP template models in the local repository. Deprecated IBP template models are not removed from the local repository, but they cannot be used to generate IBP models.

3.3. The presentation layer

The presentation layer refers to client applications that provide the application's user interface that enables organizations to access and interact with the distributed repository and make use of its functionalities.

In case the distributed repository architecture is deployed upon a logically distributed repository, a same Web application is provided to access the global repository and their local repository through Internet. Instead, in case the distributed repository architecture is deployed upon a physically distributed repository, a Web application is provided to access the global repository through Internet, and another Web application is provided for each organization to access its local repository through Internet/Intranet. Also, organizations can implement GUI-based client applications for accessing their local repository.

Other client applications can also interact with the distributed repository, such as process modeling tools [34], or software agents. An organization may delegate in software agents the responsibilities to establish dynamic agreements with other organizations [38] to collaborate and execute CBPs by creating, searching, selecting and retrieving cross-organizational collaborations and CBP models in the global repository.

4. An implementation example

The aim of this section is to describe the functionality, applicability, feasibility and utility of the proposed distributed business process model repository with an implementation of a case from a distribution network of electronic products.

4.1. Introduction

The distribution network consists of: *Megatronic*, a retailer with points of sale around the center and east regions of Argentina; and *Philkaw*, *Grundrive* and *Sanx*, which are assemblers of electronic products and suppliers of the retailer. In this network retailer and suppliers collaborate to improve their performance and achieve a high service level for final consumers. Suppliers collaborate with the retailer in a separate and peer-to-peer way. Each supplier established an independent cross-organizational collaboration with the retailer to carry out a particular collaborative business model. Megatronic agreed to carry out a CPFR model with Philkaw and Grundrive, and a VMI model with Sanx. CPFR (Collaborative Planning, Forecasting and Replenishment) [29] and VMI (Vendor Managed Inventory) [39] are collaborative business models to build demand-driven supply chains. They are oriented to align business goals and coordinate supply chain operations between suppliers and customers.

The distributed repository was implemented to allow organizations to define a collaborative network and manage the respective business process models.

4.2. Implementation and deployment of the distributed repository

The distributed repository architecture was implemented and deployed as a physically distributed repository. Components of the global repository were deployed in a public server accessible through Internet. The global repository is provided by our research group performing the role of a neutral third party. Components of local repositories were implemented and deployed by the retailer and suppliers into private servers, which are accessible through their Intranets. In this way, organizations access the global repository to manage CBP models and their local repository to manage IBP models. All the repository's functionalities described in previous sections were implemented.

The distributed repository was developed by using Eclipse, Web Services and J2EE technologies. A Web application was developed for the public global repository and another one for the private local repositories. These applications provide a Web-based user interface that enables organizations to use the repository. The global and local data repositories that kept the entities managed by the distributed repository were implemented in relational databases. The Eclipse Modeling Framework (EMF) [40] is used to serialize process models into XML files based-on the XMI format. Hence, CBP and IBP models are persisted in relational databases as serialized objects.

The implemented distributed repository supports the following process modeling languages: UP-ColBPIP for CBP models and BPMN for IBP models. Therefore, to create and edit EMF-based CBP models, the Eclipse-based tool for UP-ColBPIP [34] can be used, which supports the visual modeling of CBP models by using the UP-ColBPIP language. EMF-based BPMN models corresponding to IBP template models, which are automatically generated by the repository, are based on the BPMN 2.0 EMF meta model proposed by the BPMN 2.0 Modeler Eclipse open source project [41]. The BPMN 2.0 Modeler can be used for editing them and generating the EMF-based IBP models.

Global and local services of the repository are provided as WSDL-based Web Services and they were implemented with the J2EE technology. The implementer of the `IBPTemplateGeneration` Web service contains a model transformation engine that implements the transformation process and rules of the MDA-based method for generating IBP template models [24]. The engine was built with ATL [42,43] and it takes as input an EMF-based CBP model and generates an EMF-based BPMN model corresponding to the IBP template model of an organization.

4.3. Using the distributed repository

For using the distributed repository, the organizations have to join it by creating their profile. Each organization records the users that can access the repository on behalf of it. An organization can create a collaborative network and invite other members of the repository. In this case, the retailer created the network *Electronic Products Collaborative Distribution* and then looked for suppliers in the repository and sent them an invitation to join this network. Therefore, the suppliers joined in the collaborative network created by the retailer (Fig. 7).

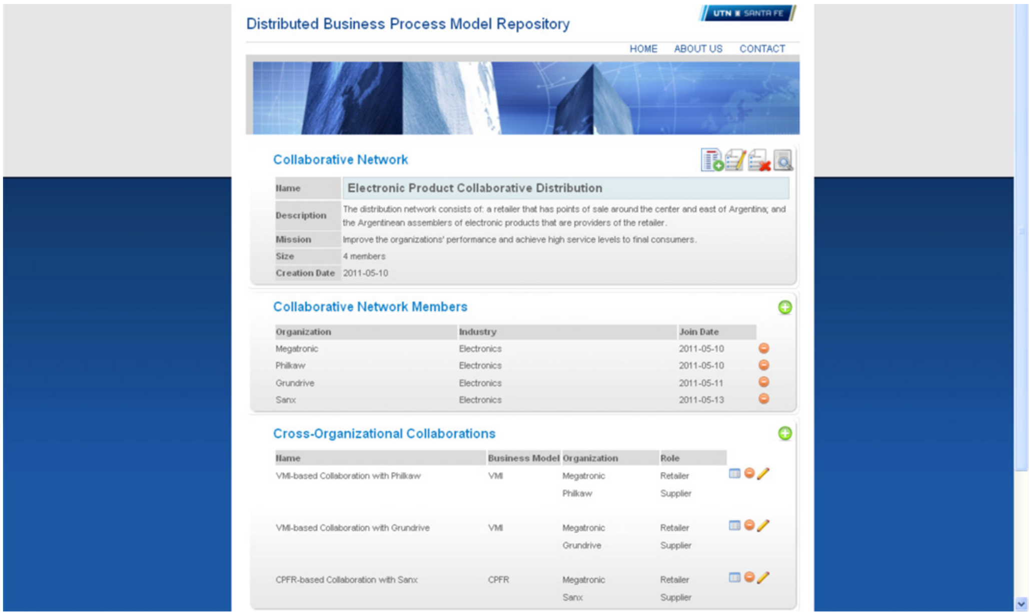


Fig. 7. Collaborative network page with information about a particular network.

As part of the network, the retailer added a catalog of CBPs, called *CPFR processes*, to include reference CBP models into the network. Two reference CBP models are available into the catalog: *Collaborative Demand Forecast Management* and *Collaborative Replenishment Plan Management*.

By using the distributed repository, three cross-organizational collaborations were defined as part of this collaborative network (Fig. 7). They refer to collaborations defined by the retailer with each supplier. Each collaboration shows the business model adopted and the role the organizations fulfill. Details about the agreement and business goals defined were also added to the collaborations.

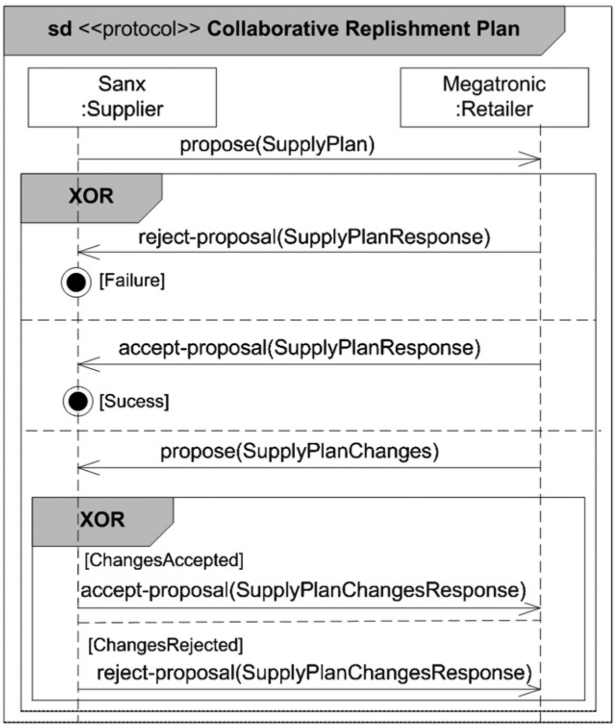


Fig. 8. *Collaborative Replenishment Plan* CBP model.

When an organization is added into a cross-organizational collaboration, the information of the collaboration is replicated in its local repository.

For each one of the created collaborations, organizations manage the corresponding CBP models. For example, the management of processes corresponding to the collaboration defined between Megatronic and Sanx is described. In this collaboration, according to the reference CPFR model adopted, the following CBPs were defined: *Demand Forecast Management*, *Replenishment Plan Management*, *Order Generation* and *Order Fulfillment*. A first version of UP-ColBPIP-based models of these processes was also added. Fig. 8 shows the interaction protocol representing the behavior of the *Replenishment Plan* CBP, which is based on the *Collaborative Replenishment Plan Management* CBP kept into the catalog of reference CBP models. This protocol manages a simple negotiation process between the retailer and the supplier for agreeing on a replenishment plan of several products for a short-time period. The protocol starts with the supplier that proposes a supply plan to the retailer, who evaluates it and decides to reject, accept or make a counter-proposal. The decision is sent to the supplier. In case of rejection or acceptation, the process finishes. In case of a counter-proposal, the supplier evaluates it and responds to the retailer with an acceptation or rejection.

When a version of a CBP model is added, it fires the sequence of interactions between services of the global repository and the local repositories of the involved parties, as it was described in Section 3.2. For example, when the first version of the *Replenishment Plan* CBP model was added, it was verified by using the `CBPmodelCorrectnessChecking` service. Because the verification process of the service returned ok, the `GlobalSynchronization` service was invoked and it fired the process that generates the IBP template models for each organization involved in the CBP. These IBP template models were stored in each of the local repositories. Fig. 9 shows the generated IBP template model for the retailer. Fig. 10 shows a screen with the resulting information viewed by the retailer in its local repository about the collaboration established with the supplier and the retailer's IBP templates, which were derived from the CBPs defined in the global repository.

From the generated IBP template models, organizations define their corresponding IBP models and add them to their local

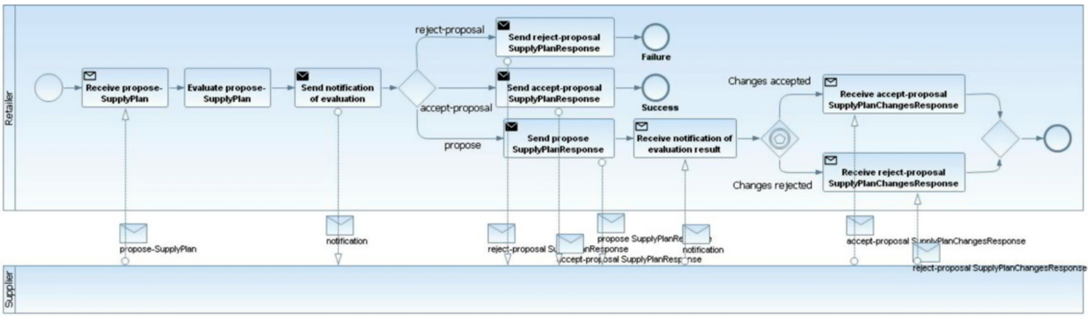


Fig. 9. Retailer's IBP template model.

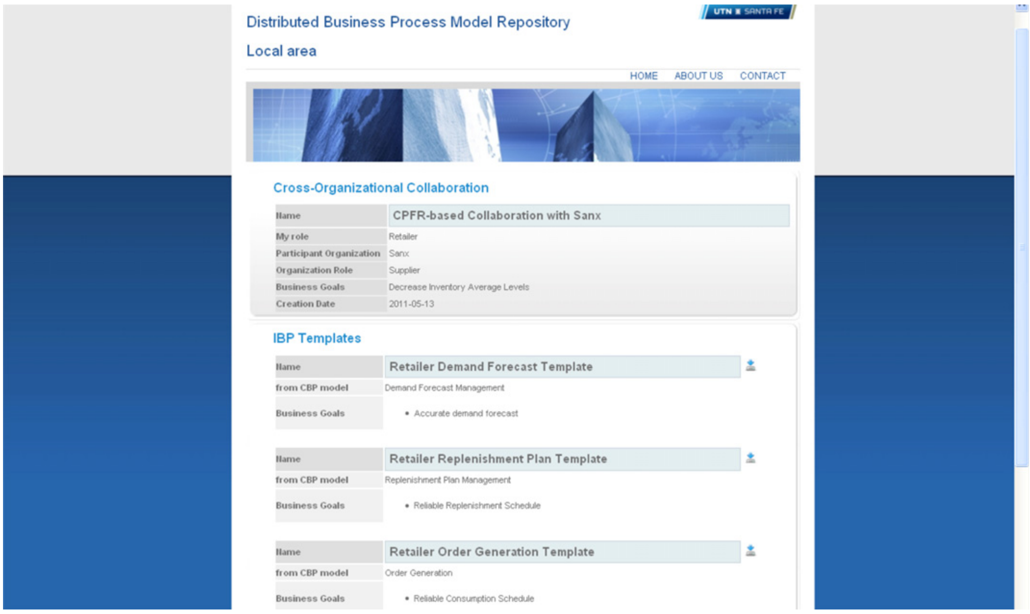


Fig. 10. Local repository page with information about a cross-organizational collaboration.

repository. For example, the retailer defined a *Retailer Replenishment Plan* IBP model replacing abstract tasks generated in the template by concrete tasks (Fig. 11). The consistency between the generated IBP template models and the defined IBP models was checked in each local repository by using the `IBPModelConsistencyChecking` service.

In this way, by using the distributed repository, each supplier and the retailer could jointly maintain and share in the global repository details of the cross-organizational collaboration and the CBP models they agreed on. In addition, by using the distributed functionality of the repository, they can maintain (in their local repositories) the particular requirements they have to fulfill to implement the CBPs agreed. These requirements are mainly expressed in the IBP models they defined. The local repositories guaranteed that these IBP models are consistent with the behavior defined in the CBP models.

As another important functionality of the distributed repository, it is described a case that required a synchronization of versions of a CBP model in the global repository with the corresponding IBP models of the organizations in their local repositories. The case refers to a new version of the *Replenishment Plan* CBP agreed by the retailer with a supplier. In this second version, the change consisted in allowing the parties to carry out more than on cycle of negotiation about a replenishment plan, which is expressed in the control flow segment `loop` of the interaction protocol of Fig. 12. When the new model version was

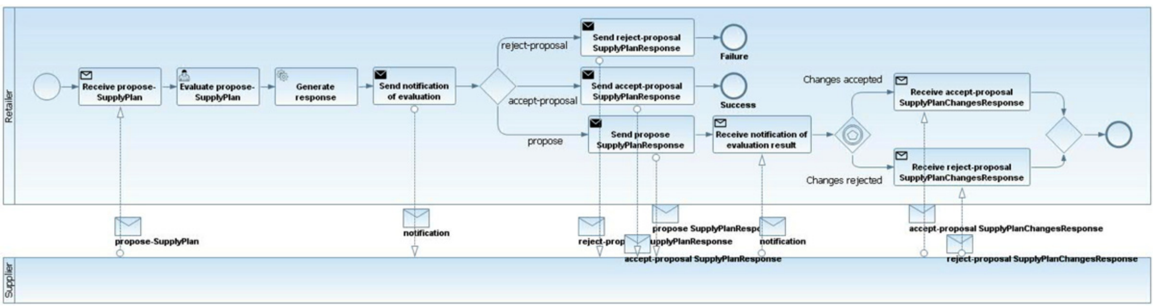


Fig. 11. Retailer's IBP model.

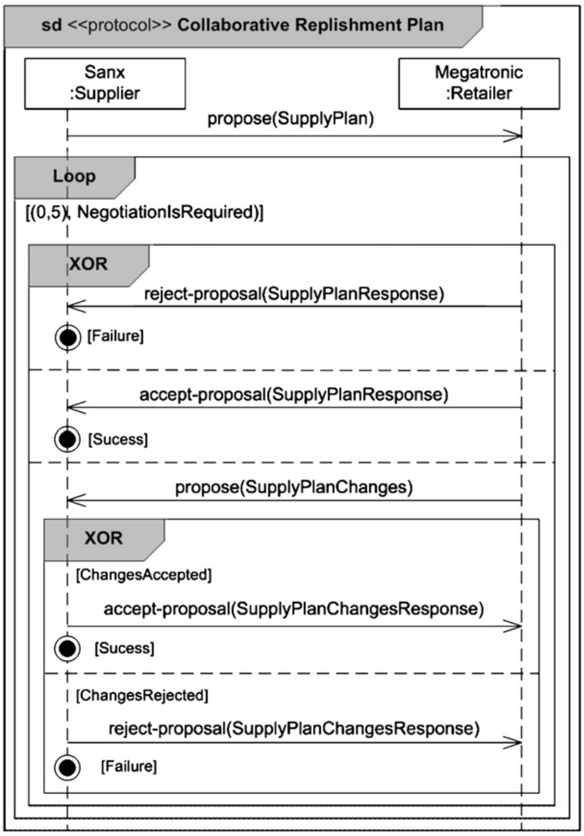


Fig. 12. New version of the *Collaborative Replenishment Plan* CBP model.

added to the *Replenishment Plan* CBP, the distributed repository made use of the `GlobalSynchronization` service to guarantee the notification of the new CBP model version and its synchronization with the IBP model versions maintained in the local repositories. The synchronization required an *IBP template model* generated for each organization and sent to the corresponding local repositories. Then, as it was proceeded previously, organizations created from the IBP template models their IBP models and stored them in their local repository. Thus, by using this distributed functionality of the repository, organizations could maintain synchronized the last version of the CBP models and their IBP models.

5. Conclusions

This section concludes the article by summarizing the contributions, the implications, the limitations, and by giving an outlook on future research.

5.1. Contributions

This article presented a distributed business process model repository for cross-organizational collaborations. The distributed repository was designed following the principles of service-oriented architectures with the aim of providing distributed, interoperable, reusable and loose-coupled services. The defined architecture enables organizations to access a global repository for managing collaborative networks, cross-organizational collaborations and CBP models. Also, it enables each organization to implement local repositories of IBP models and maintain them synchronized and consistent with CBP models of the global repository, while preserving the private aspects of the organizations.

The distributed repository aids business analysts in the design of CBP models and IBP models, and provides services to ensure these models are well-defined and consistent, guaranteeing their interoperability and keeping them synchronized. Well-defined CBP and IBP models, interoperability between IBP models, and consistency between CBP and IBP models are achieved by implementing and integrating into the repository an MDA-based method and verification methods for these process models. The design support is achieved through catalogs of reference CBP models maintained in the global repository, which can be reused by the organizations to define CBP models. In addition, design support is offered by providing a service that automatically generates IBP template models of the organizations from a CBP model. This template can be used by organizations as a blueprint to define IBP models.

A real case study presented along with a prototype implementation of the distributed repository allowed to evaluate and prove its functionality, applicability, feasibility and utility.

5.2. Implications for research and practice

Results of this paper have implications for future researches on managing large collections of business process models. While there are other proposals of business process model repositories, which vary with respect to the management techniques and the storage facilities, they do not provide functionalities to meet the identified requirements for managing process models in cross-organizational collaborations: interoperability, consistency, synchronization, privacy, and distributed implementation. The new techniques and functionalities provided by the proposed distributed repository can be took into account to improve and extend process model repositories to be used for cross-organizational collaborations.

Other researches can be oriented to use this distributed repository to observe the organizations' behaviors in other real cases and evaluate them in order to identify new requirements. Although the functionalities of the distributed repository presented in this work were evaluated through some case studies, we are aware that other requirements may need to be supported.

A number of implications related to process management practice are also identified. Most notably, the research carried out confirmed the belief that the use of a distributed repository facilities and makes it more efficient the development and management of cross-organizational collaborations. It is achieved by providing to the organizations the support to create collaborative networks, establish cross-organizational collaborations and define the CBP and IBP models, keeping them synchronized, interoperable and consistent.

Furthermore, the use of the distributed repository rises the quality of process models by providing services to maintain well-defined CBP and IBP models, decreasing costs and development time.

5.3. Limitations of the research

Firstly, the distributed repository does not provide support for editing CBP and IBP models. It also lacks support to the collaborative modeling of these process models. It only enables to download models, edit them with an external modeling tool, and upload them in the corresponding repository (either global or local) by creating a new model version.

Secondly, the distributed repository only supports the languages UP-ColBPIP for CBP models and BPMN for IBP models. This may limit the use of the repository by organizations that prefer to use other modeling languages.

5.4. Outlook

Future work is about the extension of the repository to manage system models and technological process specifications to support the phases of a Model-Driven Development Methodology for Cross-Organizational Collaborations [16]. Another work is to extend the global repository to enable organizations to manage CBP models defined with modeling languages such as Let's Dance, BPMN or BPEL4Chor. Another work is to improve some services to enable organizations to retrieve business process models that closely resemble a given business process model in order to reuse them. Another future work is to extend the distributed repository to support collaborative design of CBP models, so stakeholders distributed across the organizations can work closely to carry out the design task. The use of collaborative modeling tools, such as the presented in [44], decreases the modeling time and rises the model quality, as it was demonstrated in [45].

References

[1] G. Jiménez, N. Galeano, T. Nájera, J. Aguirre, C. Rodríguez, A. Molina, Methodology for business model definition of collaborative networked organizations, IFIP Advances in Information and Communication Technology (AICT) 186 (2005) 347–354.
[2] L. Camarinha-Matos, H. Afsarmanesh, N. Galeano, A. Molina, Collaborative networked organizations-concepts and practice in manufacturing enterprises, Computers & Industrial Engineering 57 (1) (2009) 46–60.
[3] C.-M. Chituc, C. Toscano, A.L. Azevedo, Interoperability in collaborative networks: independent and industry-specific initiatives – the case of the footwear industry, Computers in Industry 59 (7) (2008) 741–757.
[4] C.-M. Chituc, A.L. Azevedo, C. Toscano, A framework proposal for seamless interoperability in a collaborative networked environment, Computers in Industry 60 (5) (2009) 317–338.
[5] Q. Li, J. Zhou, Q. Peng, C. Li, C. Wang, J. Wu, B. Shao, Business processes oriented heterogeneous systems integration platform for networked enterprises, Computers in Industry 61 (2) (2010) 127–144.
[6] M. Weske, Business Process Management: Concepts, Languages, Architectures, Springer-Verlag Inc., New York, 2007.
[7] OMG, Business Process Model and Notation (BPMN), Version 2.0. http://www.omg.org/spec/BPMN/2.0/, 2011.
[8] B. Hofreiter, Registering UML models for global and local choreographies, in: International Conference on Electronic Commerce, vol. 342, ACM, 2008.
[9] P.D. Villarreal, E. Salomone, O. Chiotti, Modeling and specifications of collaborative business processes using a MDA approach and a UML profile, in: Enterprise Modeling and Computing with UML, Idea Group Inc., Hershey, PA, USA, 2007pp. 13–45.
[10] S. Roser, B. Bauer, A categorization of collaborative business process modeling techniques, in: International Conference on E-Commerce Technology Workshops, IEEE Computer Society, (2005), pp. 43–51.
[11] G. Decker, M. Weske, Behavioral consistency for B2B process integration, in: International Conference on Advanced Information Systems Engineering (CAiSE 2007), LNCS, vol. 4495, Springer-Verlag, Trondheim, Norway, 2007, pp. 81–95.
[12] I.M. Lazarte, O. Chiotti, P.D. Villarreal, Transforming collaborative process models into interface process models by applying an MDA approach, in: IFIP Conference on Software Services for e-Business and e-Society (I3E 2009), vol. 305, Nancy, France, (2009), pp. 301–315.
[13] B. Bauer, S. Roser, J. Müller, Adaptive design of cross-organizational business processes using a model-driven architecture, in: Wirtschaftsinformatik 2005, Springer, (2005), pp. 103–121.
[14] OASIS, WS-BPEL V2.0. www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm, 2007.
[15] M. Dumas, W. van der Aalst, A. Ter Hofstede, Process-aware Information Systems: Bridging People and Software Through Process Technology, John Wiley and Sons, Hoboken, NJ, USA, 2005.
[16] I.M. Lazarte, E. Tello-Leal, J. Roa, O. Chiotti, P.D. Villarreal, Model-driven development methodology for B2B collaborations, in: Enterprise Distributed Object Computing Conference Workshops (EDOCW), IEEE Computer Society, Vitoria, Brazil, (2010), pp. 69–78.
[17] R. Dijkman, M. Rosa, H. Reijers, Managing large collections of business process models – current techniques and challenges, Computers in Industry 63 (2) (2012) 91–97.
[18] M. La Rosa, H. Reijers, W. Aalst, R. Dijkman, J. Mendling, M. Dumas, L. Garcia-Banuelos, APROMORE: an advanced process model repository, in: Expert Systems with Applications (ESWA), vol. 38, 2011, 7029–7040.
[19] Z. Ma, B. Wetzstein, D. Anicic, S. Heymans, Semantic business process repository, in: International Workshop on Semantic Business Process Management (SBPM 2007), vol. 251, CEUR-WS.org, 2007, 92–100.
[20] J. Vanhatalo, J. Koehler, F. Leymann, Repository for business processes and arbitrary associated metadata, in: International Conference on Business Process Management, (BPM 2006), 2006, 25–31.
[21] T. Theling, J. Zwicker, P. Loos, D. Vanderhaeghen, An architecture for collaborative scenarios applying a common BPMN-repository, in: Distributed Applications and Interoperable Systems (DAIS 2005), LNCS, vol. 3543, Springer, Athens, Greece, 2005, pp. 169–180.
[22] J. Roa, O. Chiotti, P.D. Villarreal, A verification method for collaborative business processes, in: BPM 2011 International Workshops – Part I, LNBIP, vol. 99, Springer, Clermont-Ferrand, France, 2012, pp. 293–305.
[23] A. Martens, Consistency between executable and abstract processes, in: International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), IEEE Computer Society, (2005), pp. 60–67.
[24] I.M. Lazarte, P.D. Villarreal, O. Chiotti, L.H. Thom, C. Iochpe, An MDA-based method for designing integration process models in B2B collaborations, in: International Conference on Enterprise Information Systems (ICEIS 2011), INSTICC, vol. 3, SciTePress, Beijing, China, 2011, pp. 55–65.
[25] Z. Yan, R. Dijkman, P. Grefen, Business process model repositories – framework and survey, Information and Software Technology 54 (2012) 380–395.
[26] M. La Rosa, M. Dumas, R. Uba, R. Dijkman, Business process model merging: an approach to business process consolidation, ACM Transactions on Software Engineering and Methodology (TOSEM), in press.
[27] OASIS, Reference Model for SOA 1.0. docs.oasis-open.org/soa-rm/v1.0/, 2006.
[28] T. Erl, Service-oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, Boston, MA, 2005.
[29] VICS, Collaborative Planning, Forecasting and Replenishment (CPFR). www.vics.org/committees/cpfr/, 2004.
[30] OAGi, Business Objects Documents (BODs) Version 9.0. www.oagi.org/, 2009.
[31] OAGi, Universal Business Language (UBL) Version 2.0. http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html, 2006.
[32] P. Bernstein, U. Dayal, An overview of repository technology, in: International Conference on Very Large Data Bases, IEEE Computer Society, (1994), p. 705.
[33] OMG, MDA Guide Version 1.0.1. www.omg.org/cgi-bin/doc?omg/03-06-01, 2003.
[34] P.D. Villarreal, I.M. Lazarte, J. Roa, O. Chiotti, A modeling approach for collaborative business processes based on the UP-ColBPIP language, in: Business Process Management Workshops (BPM 2009), LNBIP, vol. 43, Springer, Ulm, Germany, 2010, pp. 318–329.
[35] L. Thom, M. Reichert, C. Iochpe, Activity patterns in process-aware information systems: basic concepts and empirical evidence, International Journal of Business Process Integration and Management 4 (2009) 93–110.
[36] R. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in bpmn, Information and Software Technology 50 (12) (2008) 1281–1294.
[37] A. Martens, S. Moser, A. Gerhardt, K. Funk, Analyzing compatibility of BPEL processes, in: Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), IEEE Computer Society, (2006), p. 147.
[38] E. Tello-Leal, O. Chiotti, P.D. Villarreal, Agents for managing business-to-business interactions, in: International Conference on Agents and Artificial Intelligence (ICAART 2011), Rome, Italy, 2011, 238–244.
[39] P. Franke, Vendor-managed Inventory for High Value Parts: Results from a Survey Among Leading International Manufacturing Firms, vol. 3, Univ.-Verl. der TU, Berlin, Germany, 2010.
[40] D. Steinberg, F. Budinsky, M. Paternostro, E. Merks, EMF: Eclipse Modeling Framework, Addison-Wesley Professional, Boston, MA, 2008.
[41] E. Foundation, BPMN 2.0 Modeler. www.eclipse.org/proposals/soa.bpmn2-modeler/, 2011.
[42] F. Jouault, I. Kurtev, Transforming models with ATL, in: Satellite Events at the MoDELS 2005 Conference, LNCS, vol. 3844, Springer, Montego Bay, Jamaica, 2006, pp. 128–138.
[43] F. Jouault, F. Allilaire, J. Bézivin, I. Kurtev, Atl: a model transformation tool, Science of Computer Programming 72 (1–2) (2008) 31–39.
[44] R. Brown, J. Recker, S. West, Using virtual worlds for collaborative business process modeling, Business Process Management Journal 17 (3) (2011) 546–564.
[45] P. Rittgen, IT support in collaborative modelling of business processes – a comparative experiment, International Journal of Organisational Design and Engineering 1 (1) (2010) 98–108.

Ivanna Lazarte is a PhD student in Information Systems Engineering at UTN. She has been working in the CIDISI since 2008. Her current research interest includes business process management, software engineering, model-driven development, service-oriented architectures.

Lucinéia Heloisa Thom is Post-doc researcher in Institute of Informatics at Federal University of Rio Grande do Sul since 2010. From 2010 to 2011 she was visiting researcher at University Joseph Fourier (Grenoble, France). From 2007 to 2009 she was post-doc researcher in Institute of Database and Information Systems at Ulm University working with Prof. Manfred Reichert. She received a Bachelor degree in Computer Science from the University of Santa Cruz do Sul, Brazil and a Master degree in Computer Science from the Federal University of Rio Grande do Sul (UFRGS), Brazil,

in 1999 and 2002, respectively. In 2006 she received her PhD in Computer Science from UFRGS. From 2004 to 2005 she developed part of her thesis research abroad in Institute for Parallel and Distributed Systems at University of Stuttgart. Her research interests are in the area of process management, workflow patterns, process mining, healthcare information systems and ontology. In these areas she has many published papers.

Cirano Iochpe received a PhD in Computer Science from University of Karlsruhe (Germany) and a Diploma in Data Processing from Federal University of Rio Grande do Sul (UFRGS), Brazil. He had also worked as consultant at the Data Processing Company of Porto Alegre (PROCEMPA). Besides he works as Associate Professor at UFRGS. Cirano was also Leader of several research projects in the areas of BPM and Geographical Information Systems. Related to these areas he published several papers. In 2008 he received the UTC APEX award by the PLC Network for Social and Digital Inclusion project developed in Procempa.

Omar Chiotti received a PhD from Universidad Nacional del Litoral (UNL) in 1989. He has been working for the Argentina's CONICET as a researcher since 1991. He is a Professor of Information Systems Engineering at UTN since 1986. Currently, he is the director of CIDISI, Research Center in Information Systems Engineering. His current research interest includes e-collaboration, knowledge management and multi-agent systems.

Pablo Villarreal holds a Bachelor degree and a PhD degree in Information Systems Engineering from the UTN. He is an Assistant Professor at the UTN and a Researcher at the CONICET. He works at the CIDISI since 2000. His current research interest includes business process management, workflow management, service oriented architectures, model-driven development, B2B systems, e-supply chain management and multi-agent systems.