Jörg Ziemann, Thomas Matheis, Jörn Freiheit

Modelling of Cross-Organizational Business Processes

Current Methods and Standards

Not only since the upcoming of Service-oriented Architectures the modelling of cross-organizational business processes is a heavily investigated field comprising dozens of standards based on different concepts. New techniques on the implementation site, e.g. Web Service orchestration and choreography, further extended the possibilities and requirements on such standards. To systematically order and present a comprehensive state of the art of relevant methods and standards this paper first describes requirements that occur in cross-organizational business processes both for concepts and modelling languages. Then the most important state of the art concepts for modelling cross-organizational processes are described. Based on the requirements defined before and the presented concepts, selected modelling languages are analysed.

1 Introduction

Enterprises as well as public administrations today are confronted with a highly competitive global and fast changing business environment resulting in an increasing level of cooperation between organizations. This leads to the necessity of implementing interoperable software systems and an efficient modelling of cross-organizational business processes. In the past years research presented various new concepts, standards and tools to support cross-organizational business processes (CBP). The aim of this paper is to provide stakeholders with a comprehensive overview of requirements as well as current concepts and standards for CBP modelling.

A business process is a continuous series of organizational tasks, undertaken for the purpose of creating output. While intra-organizational processes comprise activities executed inside one organization only, the activities comprised in a cross-organizational business process are executed by different organizations that are working together to reach a common objective. Business process models are developed for the purpose of documentation, optimization and automation of business processes. Though models for representing CBPs share these objectives, CBP models differ in various aspects from those for intra-organization processes, e.g. need for information hiding, (higher) need for unambiguous descriptions, need for model usability focused on the collaboration partner and support of flexible relationships.

To explain specifics of CBP modelling and to provide a basis for judging the subsequently described concepts and languages for CBP modelling, section 2 discusses the requirements on collaborative business processes. Besides general requirements, specific requirements on modelling languages are discussed that apply particularly for collaborative business process modelling. Section 3 describes existing concepts that are applicable for CBP modelling. The concepts presented there are also taken from recent literature and research projects as well as tools applied already in industry. In section 4, a selection of currently prominent modelling languages is evaluated both regarding their compliance with the previously described requirements and concepts for modelling of CBPs.

2 Requirements for modelling of CBP

In general, modelling languages have to fulfil a set of requirements like flexibility, learnability, good visualization, extensibility, display hierarchy or different levels of granularity, high expressability, executability and analyzability. For an exhaustive list of such generic requirements cp. also Frank und van Laak [FL03]. Based on a literature review (cp. e.g. [WB04]) as well as the results of national and international research projects[1] in the field of cross-organizational business process modelling we deduce the following specific requirements for CBP modelling languages. The requirements will be used later to compare the selected modelling languages.

Requirement 1: Keep private information private. Since it can not be expected that each partner publishes its entire workflow and all contained information, this requirement is essential. To allow for publishing the relevant information only, the boundary of the collaboration sphere has to be defined. Various concepts support this, including the specification of static and dynamic system interfaces and the distinction between various levels of privacy/visibility of information.

Requirement 2: Specify the interfaces of the partners formally. It is important that the interaction description of each organization involved in the CBP is unambiguous. Often, especially in larger scenarios, an automated verification of the validity of interaction sequences is necessary. To make verification possible the causal relationship at which point in time or after which event information is required, the type of the required information (e.g. document, message) and the number or amount of required information items should be specified precisely.

Requirement 3: Mapping the CBP to executable processes. The distributed execution of a CBP starts with a common process model that all partners share and that is business oriented. From this model every partner extracts those parts that he has to execute and augments them with arbitrary information he needs for execution, e.g., refinements of process sub-parts or execution context parameters [Gr06]. Thus the used modelling language should be able to transfer the CBP from business level into an IT-oriented workflow model on technical level like e.g. BPEL.

Requirement 4: Support of the data flow. It is important that the data flow of the involved partners of a CBP can be represented by the modelling language [KKS04]. Especially a description of the input needed from partners in order to execute their process parts is necessary.

Requirement 5: Support of organizational units and roles. Because different partners are involved in a CBP, it is important to describe the organizational units with the communication and reporting relationships within the CBP. Furthermore, the role concept defines the requirements profile of an organizational unit, particularly necessary for workflow applications. The term “role” describes a certain type of organizational unit with clearly defined qualifications and skills. Thus, the modelling language should be able to describe the different organizational units and roles of the partners within the CBP [KKS04]. The definition of roles also offers to associate activities with roles. Moreover, this association can further be managed by introducing, for example, separation of duties and the management of roles can further be managed by, for example, introducing delegation and revocation of rights and duties.

Requirement 6: Support of the analysis of the CBP. Collaborative business process analysis denotes all actions that aim towards measurement and examination of running and finished collaborative processes with the goal of discovering optimization potentials. Once found, such a potential can be realized by changing the process model in the modelling phase of the next cycle pass. Thus, the modelling language should support the analysis of a CBP [MSW06].

Requirement 7: Support of semantic annotation. Due to an increased risk of misunderstandings in the context of CBP modelling, precisely defined, unambiguous models are required. This comprises a common set (dictionary) of terminologies, a set of relations between terms and their transformations to private processes or terminologies. Such annotations also support transformations of process descriptions from one language into another [HW06], which often is necessary in CBP scenarios.

3 Concepts to support CBP modelling

In the previous section requirements for collaborative modelling have been discussed. In this section we present concepts that have been developed for supporting CBP modelling. Some of these concepts ease the modelling (e.g. interaction patterns), others are directly focused on CBP modelling requirements (e.g. the concept of public, private and global views aims to keep private information private). In Section 4 we then discuss some of the most important modelling languages for CBPs and check both, if they fulfil the requirements presented in Section 2 and which of them support which concepts presented in this section.

An increased competition forces organizations to concentrate on core competencies and to collaborate closely yet flexibly with other organizations. This is true not only for enterprises, but also for public ad-

[1] E.g. ATHENA IP (http://www.athena-ip.org/),
ArKos (http://www.arkos.info/),
R4eGov IP (http://www.r4egov.eu/),
Interop NOE (http://www.interop-noe.org/).

ministrations. Thus, methods are required to describe and automate cross-organizational business processes in an efficient manner. In the last decade, the area of cross-organizational processes was investigated intensively, e.g. Van der Aalst[2], Petri Nets[3], workflow research[4] and has been studied by various research Projects, e.g. ArKOS[5], ATHENA[6] and Interop NOE[7].

Van der Aalst [Va99] reviewed various forms of interoperability and their usefulness in the context of E-commerce. There he identified the following five forms of realizing interoperable systems by enacting cross-organizational business processes: Capacity sharing, Subcontracting, Chained execution, Case transfer, Loosely coupled. For facilitating the modelling of CBP, van der Aalst also described the Public-To-Private (P2P) approach, which provides the means to specify a common public workflow, to partition it according to the organizations involved and to allow for private refinement of the parts by the organizations, based on a notion of inheritance [VW01]. The P2P approach guarantees that the private workflows of the participating organizations satisfy the public workflow as agreed upon. He also described a top-down (or "outside-in") approach to come from global processes to private processes.

Schulz and Orlowska's model is different from 1-tier peer-to-peer model and 2-tier private-public model. They stress that the proposed model framework keeps a minimal set of workflow-relevant data and protocol information, in such a way the workflows can be reused and their privacy be maintained.

Greiner et al.'s work [Gr06] describes the designing and implementing of cross-organizational business processes including different levels of technical detail: the business level, technical level and execution level. They identify how the mappings and transformations are needed among private process, view process and "global" process among the different levels. The business level models illustrate the organizational business aspects as a prerequisite for the successful technical integration of IT systems or their configurations. The technical model derived from the business level model secures the technical realization of the process integration and represents the bridge to the process execution.

Recent research efforts in CBP design were accompanied by the upcoming of new SOA standards and related concepts like Web Service Choreography and Web Service Orchestration. As an outcome, three major types of models supplementing cross-organizational businesses can be observed: First, a "big picture" of the overall CBP, also called global process model, that displays all organizations involved in the CBP and their interactions. Second, models of so called private processes which are executed inside an individual organization and should not be published (completely) to collaboration partners; nonetheless, they contain some activities that contribute to the CBP. And third, public processes – also called business process stubs – that display only those parts of the private processes relevant for the interaction with the other organizations. The same model types are, however, described differently depending on the research community that uses the model type. For example, Schulz and Orlowska [SO02] also proposed a 3-tier workflow model for cross organizational workflow that captures private partner workflows, workflow-views and coalition workflows. In the following, we shortly describe 7 concepts that in our perception represent the most relevant concepts to be captured in CBP design.

3.1 Distinguishing between models for private, public and global processes

As mentioned before, to describe and automate collaborative processes in the last years three different types of process models were introduced [Gr06] [An03]: Private, public and global process models. A **private process** model is described from the viewpoint of an individual organization. Though it may contain activities that represent interactions with other organizations, it is developed for internal use and thus may contain confidential information to be hidden from other organizations. A **public process** model is also described from the viewpoint of an individual organization. It describes the interaction of one organization (e.g. Organization A) with one (B) or more (C) partner organizations. It describes all activities of A being part of this interaction (e.g. "Send RFQ Message to B", "Receive Quote Message from C") and the causality of these activities. One way to create a public process is to derive it from a corresponding private process by abstracting all information from it that should be hidden from partner organizations. A **global process** model describes interactions between two or more organizations from a global view point [KL03]. It captures all allowed interactions between all partners involved in the collaboration. Thus, while the public process of A only captures the interactions between the organizations A and B as well as the interactions between A and C, a global process model could contain additionally the interactions between B and C.

While more technical definitions [Bu02] of public processes focus on digital message exchanges, on a

[2] http://is.tm.tue.nl/staff/wvdaalst/
[3] http://www.informatik.uni-hamburg.de/TGI/PetriNets/
[4] http://www.workflow-research.de/
[5] http://www.arkos.info/
[6] http://www.athena-ip.org/
[7] http://www.interop-noe.org/

more conceptual level interaction models can also describe material exchanges as well as the place and time of such exchanges. A process can be seen as the combination of various organizational dimensions, e.g. the dimensions function, organization, data, output and control [Sc98]. A function represents a business activity, the organizational view describes departments and roles involved in the activity, data and output describe digital and material entities consumed and produced by functions and the control flow combines these views and puts the functions in a timely order. Public processes can be seen as interfaces of private processes and should contain all information necessary to enable the interaction of different private processes. Therefore, besides the sequence of functions contained in an interaction, public processes also have to display information regarding the exchanged data (e.g. which structure an exchanged message has), the goods and services exchanged as well as the organizational departments and roles involved in the interaction.

3.2 The concept of controllability

In contrast to public views that are used to offer insight into own private processes, a recent approach has been developed within the concept of controllability [Lo06]. Although this approach has been developed for services (and thus for the execution layer), it can be adapted to the conceptual layer as well. Intuitively, controllability means that a workflow can interact properly with another workflow. In order to detect controllability, a strategy for the own workflow is generated. A strategy describes a set of workflows that could interact with the own workflow. A strategy, usually, is an automaton, which then is sent to the partner. Using this automaton, the partner can check if his own workflow is a proper partner to the other one.

3.3 Dividing global process models in Swim-lanes

A swim-lane is a concept for partitioning process models in various subsets, where each subset is executed by one specific actor or organization. Swim-lanes clearly indicate who has the responsibility for carrying out a particular activity or subset of the process. Parallel lines divide the process model into lanes that group activities of the process model by resource definitions, roles, classifiers, organization units or locations. Lanes are arranged either horizontally (rows) or vertically (columns) to divide the process model into logical areas or partitions.[8] Clear distinctions can be made between the processes within an organization unit (within a lane) and those process steps where interactions occur (across lanes). Swim-lanes can contain other swim-lanes which are called child swim-lanes. From a process management perspective, swim-lanes are also used to depict ownership or responsibility of all activities and processes within those swim-lanes. There are a lot of modelling methodologies that use the concept of swim-lanes as a technique to organize activities and to structure the layout of models in order to illustrate different functional capabilities or responsibilities. Swim-lanes are used in UML activity diagrams to logically group activities that correspond to a particular object as well as in the BPMN.[9] Further on, the concept can be used in Event-driven Process Chains (EPCs) in order to divide additional information like data or responsible persons from the current process flow [KKS04]. The use of swim-lanes in the context of SAP Business Scenario Maps[10] aims to indicate how enterprises can collaborate with each other to document the added value potential using a well understandable structure. The swim-lane concept can be used to structure process models. However, the concept does not provide a methodology to model processes. Note that one swim-lane, e.g. the swim-lane for organization A, can also be interpreted as the public process of organization A, since it describes all public interactions that this organization is involved in.

3.4 Interaction patterns

The interaction and communication between different public administrations can be very complex. Even a single communication action within a collaborative process can have a great range of formats, structures and contents. Interaction patterns try to classify messages and to define typical structures based on these classifications. Most of the research that has been done for interaction patterns deals with processes on the execution layer [BDO05] [DP06]. Patterns are used for example to transform BPEL processes into Petri nets in order to analyse the Petri net models and also in order to analyse the controllability of a process.[11] Due to the transformation ability between Petri nets and BPEL, it is possible to use the same patterns that exist for BPEL also for the conceptual layer.

On the conceptual layer there exist several transformations between the different languages, e.g. between EPCs and Petri nets. Service interaction

[8] Object Management Group (OMG): BPMN 1.0 Specification, 2006, http://www.bpmn.org

[9] UML. http://www.uml.org/

[10] SAP. Sap business maps. Technical report, SAP AG, http://www.sap.com/solutions/businessmaps/c-businessmaps/, 2004.

[11] http://sky.fit.qut.edu.au/~dumas/ServiceInteractionPatterns/patterns.html

patterns have mainly been developed at the Queensland University by Barros, Dumas and ter Hofstedte [BDH05] [BDB05].

3.5 Visualization of static interfaces

The trend of cooperation intensifies the need for modelling-methods that consider explicitly the interfaces between more companies participating in one global business process. Generally, an interface, according to the DIN 44300, is defined as an intended or real crossover of the boundary between two units of a same kind respecting the agreed rules for the delivery of data or signals [DIN88]. In object-oriented approaches, the term interface denominates the totality of the public methods of an object [Ha01]. Whereas e.g. the EPC method provides edges and connectors for defining the control-flow, until now, there has been no methodical support for the representation of the crossover between single functions. If two functions that should be processed sequentially reside in two organisations that are separated from each other, the business-process rules that would ensure a smooth transfer of a control-flow from one organisation to another, have to be defined [HK02].

In practice it has been shown that first approaches to model and visualise the cooperative business-processes such as e.g. the SAP AG's C-business maps show a very high aggregation level and simply express the existence of a process-interface without providing a real technical added value. Although the necessity for introducing process interfaces, e.g. in the context of the modelling of the services, has already been detected, a detailed conceptual specification of a transfer from one process partner to another remained undone [KZ03]. A suggestion for a concrete configuration of the conceptual specification of an interface was presented by Kupsch and Klein [KKS04]. In order to get a compact, intuitively understandable visual representation of interfaces in association with e.g. EPC, an interface-diagram is recommended for its conceptual specification. It contains, depending on the company's goals that the entire process supports, substantial dimensions that are necessary for a successful performance of the processes. For interfaces of each collaborative process, an appropriate diagram is created, that is identified through the common name of the process type. The functions that precede or follow an interface make part of a partner-individual pool. In order to ensure transparency, the name of an appropriate function/process module is introduced, depending on the aggregation level applied. Each module can then be specified in various dimensions. Kupsch and Klein propose the dimensions of time, flexibility, place and output for each interaction module [KKS04].

3.6 Semantic annotation of modelling language

Many problems associated with CBP are semantic in nature, coming up when describing resources to be exchanged and knowledge to be shared. Hence, if a more automated solution is required, solutions that describe precisely the models of CBPs, resources and information are needed.

In the last years these problems have been studied also in the field of web services to support automatic discovery and composition of services. From this research, a number of results are now available which (partly) are also applicable for CBP modelling. These results are mainly based on the concepts of *reference ontology*, *semantic annotation* and *semantic services*. The basic idea is that resources must be annotated through a reference ontology, i.e. a structured glossary of concepts shared by a community. The annotated resources are stored in repositories and can be discovered through searching algorithms and composed through reconciliation procedures. In general, in CBPs two different aspects of business process models can be identified that are to be annotated semantically: Elements describing the structure of the process model, e.g. control flow elements like logical connectors, and elements describing information, material or other artefacts that are objects used by the process activities, e.g. Business documents, material that is to be sent, money that is to be received, etc. Further on, in the context of CBPs, use of semantic annotations can broadly be categorized into

- Semantic annotation for enabling **horizontal matchmaking**. Horizontal refers to the fact, that the models that are to be matched are on the same level of abstraction. For example, government agency A could provide a semantically annotated EPC model and agency B tries to match its own EPC model with the model of A.
- Semantic annotation for enabling **vertical model transformation** or synchronization. This refers to annotations aiming to describe exactly the elements of a model for connecting it with a model on a different level of (technical) abstraction. For example, the elements of an EPC could be annotated in such a way, that their relation to programming language constructs would be clear.

For the sake of horizontal matchmaking, Thomas and Fellmann [TF06] describe a concept to annotate (graphical) EPC models with graphical OWL models. Correspondingly, they describe how theses models can be described with XML representations of both languages, e.g. with EPML and RDF.

3.7 Representing long running transactions

Many real-life CBPs have high requirements regarding consistency and can be running over long periods of time. Especially in distributed environments, it is difficult to control and ensure the consistency of data belonging to one business process. To make this task easier, in recent years the concept of transactionality was taken from the field of database research and was applied to business processes. On the one hand standards for the execution of consistent transactions were created. On the other hand, the need to display secure transactions in business models was met and first modelling languages were offered that contain elements to depict transactions. In general, classical (database) transactions operate through a small period of time and they are characterised by the ACID-principle (Atomicity, Consistency, Isolation und Durability).

It attests that a sequence of activities executed on one system can be seen as a single transaction, if it satisfies following specifications: either all activities are effectually completed or they are eventually not started (Atomicity), the activities cause a consistent system state (Consistency), the activities do not effect any operations that are not part of the transaction, unless this operation is explicitly visible (Isolation), and the sequence is not cancelled by a system error after its execution (Durability) [LR97]. The most popular model for atomic behaviour implementation is the 2-Phase-Commit-Protocol (2PC). However, this includes the locking of the resources affected by the transaction, i.e. the resources are locked at the beginning of a transaction and can not be changed by another transaction until the end of the actual transaction [LR00].

Business process transactions have to be able to cover not transactional programs, long lasting activities and human activities. Such transactions are also called Long-Running Business Transactions (LRT) or Business Transactions. These transactions are supported by the Open Nested Transaction Model and also by the Sagas [GMS87] transaction model [LR97]. Within atomic transaction models (e.g. 2PC) a rollback occurs before the transaction's closure. In case a transaction should not be completed, the locking is taken from the resources. This means that the resources were not changed during the transaction process. However, a compensation action is applied to the transaction activities *after* the transaction's closure, i.e. after the resources have been changed. A compensation sphere is an activities sequence, which either has to be completely executed or completely revised (compensated). For that purpose a compensation activity is assigned either to the single activities or to the whole sphere. This compensation activity is executed, when a transaction has to be rolled back or interrupted [LR97]. Since interoperability should start on the design level and important transactions should be defined by process designers, modelling languages suitable for R4eGov should support advanced transaction mechanisms spanning various parties. BPMN[12] is one of the few languages who include this, and will be presented here as an example. For being able to represent long running business transactions, languages should be able to represent compensation spheres and corresponding compensation activities. The difficulty of this endeavour arises by fact that these elements can appear in various models and in different levels of detail. For example, the public process interfaces (e.g. 2 different models) of organization A and B can depict elements belonging to the same compensation sphere. Moreover, since transactions can be nested they have to be modelled on various process levels, e.g. in top-processes and underlying sub-processes.

4 Suitability of current standards for CBP modelling

Although there is an abundance of business process modelling languages, only a few were applicable for CBP modelling in practical cases. One major requirement of stakeholders in practice is that business process modelling languages should be widely used in industry and in commercial products. This is the case for EPCs and UML. UML is of additional importance because there is a strong organization behind UML pushing it. This is also the case for BPMN and we have therefore selected it for a more detailed introduction in this section. Another reason of importance is the ability of formal analysis, optimization and verification, which is the case for Petri nets. Thus, in this section we analyse the modelling languages UML, BPMN, EPC and Petri nets in more detail. Note that we selected those for languages from a list of 14 well known standards (cp. [FMZ07]), but due to space restrictions, we omit the other languages here.

If we consider that the requirement to **keep private information private** can be fulfilled by publishing public information of the process only, then this can be done by using a highly abstracted model of the private workflow. "Highly abstracted" in this context means private submodels, i.e. parts of the private workflow, that are just published as, for example, one activity. This is closely related to a hierarchical structure of the model, where on the top level of the hierarchy there is a very abstract presentation of the workflow including all interactions with collaborative

[12] Business Process Modelling Notation Specification, OMG 2006, http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf

workflows and on lower levels there are refinements of this abstraction. However, in order to keep private information private, only the top level needs to be published. The concept of hierarchy can be applied to all four languages. There are in particular broad theories of hierarchy for EPCs and Petri nets.

All analysed modelling languages have deficiencies regarding the **formal specification of interfaces**. Petri nets come with formal semantics. The interface of a Petri net model usually is specified by places that act as channels for message passing or document exchange. The types of data to be exchanged can formally be specified using coloured Petri nets. However, the inherently weak support of CBP typical elements (organizational roles, different document types etc.) in Petri nets decreases the expressiveness of their interfaces. UML and in particular BPMN and EPC are more powerful in terms of modelling interfaces having special modelling elements for interfaces and triggering events but are weaker regarding formal semantics.

	UML	Petri nets	BPMN	EPC
Keep private information private	+	+	+	+
Specify the interfaces formally	o	+	-	o
Mapping the CBP to executable processes	+	+	+	o
Support of data flow	o	+	o	o
Support of involved roles	-	o	-	+
Support of analysis of the CBP	-	+	-	+
Support of semantic annotation	-	-	-	O
Criterion is satisfied (+), criterion is partly considered (o), criterion is not supported (-)				

Table 1: Requirements fulfilled by selected business process modelling languages

The **mapping to executable processes** is possible for all languages. In particular, mappings from all for modelling languages to BPEL exist. However, this cannot be done purely automatic but with computer support. For BPMN a mapping to BPEL is already contained in the languages specification and BPMN elements are matching well with BPEL concepts, for example both languages support elements for distributed transactions. EPC are also a suitable basis for BPEL transformations (cp. [ZM05]) and the core elements of EPCs (functions and events) map to the core elements of BPEL (web service invocations and various types of events). Petri nets can be executed even without mapping to BPEL due to its formal semantics. The BPMN specification already contains a detailed description of transformation to BPEL.

Data flow is fully supported by Petri nets due to their ability of specifying very complex types (for coloured Petri nets) of *tokens*, i.e. the data or information are modelled explicitly and the tokens are evaluated in order to compute the occurrence of certain events. In the other languages data flow is not directly supported (no data flows through the process models) but they offer modelling elements for different types of data (data bases, documents, etc.), which can be associated with activities, such that the flow and the change of data can be modelled indirectly.

All languages focus on process activities and not on the executing actors or a detailed description of their **roles**. However, EPCs are suited here because of their ability to model organizations and to specify who is in charge of certain activities. There is no direct support of modelling resources (e.g. staff members) in Petri nets but this can be done by modelling resource places and marking them with corresponding tokens.

The ability to **analyze and optimize** CBPs requires appropriate modelling elements (e.g. performance indicators). There has been done work on EPCs and Petri nets in order to introduce performance modelling to those languages. In addition, Petri nets support modelling elements for the verification of CBPs. UML and BPMN lack comparable perfomance modelling elements.

There has been done work for **semantic annotation** for EPCs. However, to our knowledge there is no other approach on semantic annotation for the other selected languages.

For all four languages the concept of **Swim-lanes** is applicable. Though use of swim lanes is more common for BPMN and UML, they are as well applied to EPC and Petri nets. As discussed above for the requirement of keeping private information private, in general for it is possible for all process languages to derive **global and public processes from private processes**. However, to our knowledge explicit approaches for this kind of transformations exist for EPCs and Petri nets only. Concepts for **visualization of static interfaces** was described for the EPC [KKS04] and makes most sense for business oriented languages, e.g. the more formal Petri Nets are less suitable for such visualizations. Among the four languages, **semantic annotations** of business process modelling languages so far exist only for the EPC [TF06].

Long running **transactions** are supported explicitly only by BPMN. The concept of **controllability** is closely related to the concept of private and public workflows. It has been developed for Petri nets but can certainly be applied to the other languages as well. **Interaction patterns** mainly exist for Petri nets and BPEL but can certainly be developed for the other languages as well.

	UML	Petri nets	BPMN	EPC
Swim-lanes	+	o	+	o
Private, public and global processes	o	+	o	+
Visualization of static interfaces	o	-	o	+
Semantic annotation of modelling languages	-	-	-	+
Representing long running transactions	-	-	+	-
Controllability	o	+	o	o
Interaction patterns	o	+	o	O
Criterion is satisfied (+), criterion is partly considered (o), criterion is not supported (-)				

Table 2: Concepts supported by selected business process modelling languages

5 Summary and future research

To describe and analyse existing approaches to model CBPs we first described requirements distinct for cross-organizational scenarios. Then state of the art concepts for modelling the conceptual layer of collaborative processes were described, including swim-lanes, semantical enriched processes, interaction patterns, and the distinction between public, private or global views, which are supplemented by the controllability approach. The latter approach, however, is not ready for application yet and has to be tested sufficient for real processes. Moreover, it has been suggested for business process execution and not yet applied to conceptual models. Choosing from a list of 14 well known standards, we selected and analysed EPC, BPMN, UML and Petri Nets based on the gathered requirements. Due to their explicit support of business elements, EPC and BPMN seem to be the most suitable candidates for modelling CBPs. However, for BPMN an established commercial tool is missing, which do exist for EPCs (e.g. the ARIS Toolset). Petri nets are more appropriate for formal analysis of business processes. However, besides the missing commercial tool, there is also a lack of modelling power in terms of different available process elements, such as organizational diagrams, interfaces etc. Nonetheless, Petri nets have implicit formal semantics and are able to model objects flowing through a process. There exist several enhancement and transformation approaches for EPCs, e.g. EPC models can be directly transformed into executable formats, such as BPEL or enriched with semantic annotations. Thus, our future research will concentrate on applying and extending BPMN and EPC for cross-organizational business process modelling.

The work published in this paper is (partly) funded by the E.C. through the R4eGov project. It does not represent the view of E.C. or the R4eGov consortium, and authors are solely responsible for the paper's content.

References

[An03] Andrews, T.; Curbera, F.; Dholakia, H.; Goland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.; Trickovic, I.; Weerawarana, S. (2003): Business Process Execution Language for Web Services – Version 1.1. 2003.

[BDB05] Barros, A.; Dumas, M.; Bruza, P.: The Move to Web Service Ecosystems. BPTrends Newsletter, Volume 3, Number 12, December 2005.

[BDH05] Barros, A.; Dumas, M.; ter Hofstede, A.: Service Interaction Patterns: Towards a Reference Framework for Service-based Business Process Interconnection. Technical Report FIT-TR-2005-02, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, March 2005.

[BDO05] Barros, A.; Dumas, M.; Oaks, P.: Standards for Web Service Choreography and Orchestration: Issues and Perspectives. In Proceedings of the Workshop on Web Service Choreography and Orchestration for Business Process Management, Nancy, France, Springer Verlag, September 2005.

[Bu02] Bussler, C.; Public Process Inheritance for Business-to-Business Integration. In Buchmann, A. P.; Casati, F.; Fiege, L.; Shan, M. C.: Technologies for E-Services – Third International Workshop TES 2002, Hong Kong, 2002.

[DIN88] Deutsches Institut für Normung e.V.: DIN 44300 : Informationsverarbeitung : Begriffe. Teil 1. Berlin : Beuth, 1988.

[DP06] Decker, G.; Puhlmann, F.: Formalizing Service Interactions Extended version of a paper to be published in the 4th International Conference on Business Process Management (BPM'2006), Vienna, Austria, September 2006.

[FL03] Frank, U.; van Laak, B.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsbericht des Instituts für Wirtschafts- und Verwaltungsinformatik der Universität Koblenz. http://www.uni-koblenz.de/~iwi/publicfiles/Arbeitsberichte/Nr34.pdf, cited 2004-02-26.

[FMZ07] Freiheit, J.; Matheis, T.; Ziemann, J.; Definition of static and dynamic models of collaborative workflow interoperability. Deliverable D4.1, R4eGov – Towards e-Administration in the large. IST-2004-026650.

[GMS87] Garcia-Molina, H.; Salem, K.: Sagas ; in ACM SIGMOD; 1987; San Francisco; pp. 249-260, 1987.

[Gr06] Greiner, U.; Lippe, S.; Kahl, T.; Ziemann, J.; Jäkel, F.W.:Designing and implementing cross-organizational business processes – description and application of a modelling framework. In Proceedings of the Interoperability for Enterprise. Software and Applications Conference (I-ESA 2006).

[Ha01] Hansen, H. R.: Wirtschaftsinformatik I. 8. Auflage. Stuttgart : Lucius & Lucius, 2001.

[HK02] Herrmann, K.; Klein, R.: Effizientes Schnittstellenmanagement : Erfolgsfaktor für die E-Collaboration. In: IM – Information Management & Consulting Nr. 4, 2002.

[HW06] Herborn, T.; Wimmer, M.: Process Ontologies Facilitating Interoperability in eGovernment – A Methodological Framework. In: Proceeding of the Workshop on Semantics for Business Process Management. p. 76-88, 2006.

[KKS04] Klein, R.; Kupsch, F.; Scheer, A.-W.: Modellierung inter-organisationaler Prozesse mit Ereignisgesteuerten Prozessketten. In: Scheer, A.-W. (Ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 178, Saarbruecken, 2004.

[KL03] Khalaf, R.; Leymann, F.: On Web Services Aggregation. In: Benatallah, B.; Shan, M. (Eds.): Technologies for E-Services. Lecture Notes in Computer Sciences 2819, Springer, Heidelberg, 2003.

[KZ03] Klein, C.; Zürn, A.: Einsatz von Prozessmodulen im Service Engineering : Praxisbeispiel und Problemfelder. In: Bullinger, H.-J.; Scheer, A.-W. (Eds.): Service Engineering : Entwicklung und Gestaltung innovativer Dienstleistungen. Berlin [u. a.]: Springer, 2003, p. 737.

[LR00] Leymann, F.; Roller, D.: Production Workflow – Concepts and Techniques PTR Prentice Hall, 2000.

[LR97] Leymann, F.; Roller, D.: Workflow based applications, IBM Systems Journal 36(1) pp. 102-123, 1997.

[Lo06] Lohmann, N.; Massuthe P.; Stahl, Ch.; Weinberg D.: Analyzing Interacting BPEL Processes. Intern. Conf. on Business Process Modelling (BPM 2006). 2006

[MSW06] Matheis, T.; Simon, B.; Werth, D.: Process-Based Performance Measurement of Networked Businesses. In: Cunningham, P.: Cunningham, M. (Eds.): Exploiting the Knowledge Economy – Issues, Applications, Case Studies, eChallenges e-2006 Conference, Barcelona, 25.-27, pp. 20-28. (ISBN: 1-58603-682-3), Oktober 2006.

[Pe62] Petri, C. A.: Kommunikation mit Automaten. Dissertation, Technische Universität Darmstadt. Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.

[Sc98] Scheer, A.-W.; ARIS – Vom Geschäftsprozess zum Anwendungssystem. 3. edition. Springer, Berlin, 1998.

[SO02] Schultz, K.; Orlowska, M.: Towards a cross-organizational workflow model, Proc. 3rd IFIP Conf. on Infrastructures for Virtual Enterprise, May 1-3, 2002, Sesimbra, Kluwer.

[TF06] Thomas, O.; Fellmann, M.: Semantische Ereignisgesteuerte Prozessketten. In: Schelp, J.; Winter, R.; Frank, U.; Rieger, B.; Turowski, K. (Eds.): Integration, Informationslogistik und Architektur : DW2006, 21.-22. Sept. 2006, Friedrichshafen : Proceedings. Bonn : Köllen, 2006 (LNI, P-90), p. 205-224.

[Va99] Van Der Aalst: Process oriented architectures for electronic commerce and interorganizational workflow. Information Systems,24(8): pp. 639 – 671, 1999.

[VW01] Van der Aalst W. M. P.; Weske, M.: The P2P Approach to Interorganizational Workflows, Proceedings of the 13th International Conference on Advanced Information Systems Engineering, p. 140-156, 2001.

[WA04] Wombacher, A.; Aberer, K.: Requirements for Workflow Modeling in P2P-Workflows derived from Collaboration Establishment; in Proc. 1st Intl. Workshop on Business Process Integration and Management (BPIM 04), Zaragoza, Spain, IEEE Computer Cociety Press, ISBN: 0-7695-2195-9, p. 1036-1041, 2004.

[ZM05] Ziemann, J.; Mendling, J.: Transformation of EPCs to BPEL – A pragmatic approach. 7th International Conference on the Modern Information Technology in the Innovation Processes of the industrial enterprises, Genoa, Italy, 2005.

Jörg Ziemann, Thomas Matheis

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum
für Künstliche Intelligenz (DFKI)
Stuhlsatzenhausweg 3, Geb. D3 2
66123 Saarbrücken
Germany
{joerg.ziemann|thomas.matheis}@iwi.dfki.de

Jörn Freiheit

Max-Planck-Institute Saarbrücken
Stuhlsatzenhausweg 85
66121 Saarbrücken
Germany
freiheit@mpi-sb.mpg.de