Using BPEL processes defined by
Event-driven Process Chains

Carlo Simon[1], Jörn Freiheit[2] and Sebastian Olbrich[3]

(1) University Koblenz-Landau, simon@uni-koblenz.de
(2) Max-Planck Institute, Saarbrücken, freiheit@mpi-inf.mpg.de
(3) Philipps University at Marburg, sebastian.olbrich@zeon.de

Abstract: The paper discusses the usability concept of workflow services for event-driven process chains. Usability is similar to controllability, known from Workflow net based BPEL semantics theory. Based on the observation, that if business processes interact (for instance because one of them is a service), it must be possible to check whether this is structurally possible or not. In opposite to prior work, the focus is laid on the exchange of information rather than on the integration of shared events and functions. The theoretical outcome of the paper is applied to a German E-Government (web) service.

1 Introduction

The definition and acceptance of the *business process execution language for web-services* (*BPEL4WS* or *BPEL* for short, [ACD⁺03, Ju04]) by the key players in the software industry (especially in the field of workflow management systems) made this language a de-facto standard for the specification of business processes and services. BPEL, however, does not only standardise existing exchange languages but, moreover, puts a new concept into the centre of the considerations: the development of *service-oriented information systems' architectures* instead of monolithic ones. BPEL systems are highly distributed and have to interact, i.e. *communicate* with each other. With the introduction of this paradigm, also new challenges and questions concerning the modelling of such (distributed) systems came up: (1) What are appropriate modelling languages for such systems? (2) Which properties of such systems must be checked in order to validate their proper work.

One of the main difficulties in answering these questions is that the *specification of BPEL* is given in natural language, i.e. no mathematical interpretation of BPEL processes exists. Consequently, there does not exist a verification technique which can be derived from the BPEL specification to verify the collaboration of BPEL services. A possible solution to this is to define such a semantics after the event as discussed in Section 5. By introducing - for instance - a Workflow net semantics it is possible to answer the question whether a BPEL service can *be controlled* by another. An application of this approach obviously makes sense, if the BPEL specifications are *already given*.

Another possible starting point is a description of business processes which has *not already* been translated into BPEL specifications. This is, for example the case, if an organisation has to decide on shifting its information system to a *service-oriented architecture (SOA)*. At this moment, the services of this SOA must be identified, specified and their usability must be guaranteed. Since event-driven process chain (EPC) is one of the most popular business process modelling languages, it is both in theory and practice a highly relevant problem *to answer usability questions on the base of EPC models*.

The automatic synthesis of an environment for a modelled dynamic system - a business process or a machine - is a means to describe its *controllability*. This, however, demands a formal state semantics of the system to be controlled which is usually not given for a business environment. Nonetheless, many ideas and concepts of controllability - as shown in this paper - can be transferred also to conceptual process models although they cannot be mapped into a discrete finite state space. Since there are differences between the two approaches, we use the term *usability* for our findings for conceptual process models.

Although the consideration of this problem makes use of prior work on the integration of EPC models discussed in Section 4, this is not the ultimate solution. The integration described there is based on merging events and functions with similar meaning in single elements, i.e. the modelled processes are synchronised within these elements. A service, however, is decoupled from its user and they might operate asynchronously. Then a message passing mechanism is required rather than a synchronisation.

In order to explain the problem, to summarise the prior work, to develop a new theoretic result, and to apply this to a modelling problem, this paper is organised as follows: after an introduction to communicating workflow services in Section 2, prior work on giving a state semantics to BPEL processes to demonstrate their controllability is discussed in Section 3. Afterwards, prior work on the integration of EPC models is discussed in Section 4 in contrast to message passing between EPC models considered in Section 5. An application of the approach is given in Section 6 before the paper ends with some concluding remarks.

2 Communicating Workflow Services

There exist various definitions of *business processes*, that all describe business processes as interconnected activities to achieve a specific business goal [Ga83, Ja96, JB96, St01, AH02, Ga03, Ho04, SS04]. In a more specific extension of this definition, the fulfilment of customer needs is seen as the ultimate goal [Da93, HC93, Po04]. The Workflow Management Coalition (WfMC) defines in addition *workflows* as electronically supported business processes [Ho95].

Workflows are increasingly used to specify the behaviour of services and especially of web services [RSS05]. The use of services makes it possible to provide support for business processes that must be executed in a distributed, heterogeneous environment. As pointed out in [RSS05], the communication of services is no simple input/output relationship but during service execution the service has to interact with its environment. Consequently, the interface has a *life-time* and its structure might change during this time.

Currently, one of the most widely discussed languages for the specification of business processes is the *business process execution language for web services* (*BPEL4WS* or *BPEL* for short) [ACD$^+$03, Ju04]. The language is based on formerly defined languages such as IBM's WSDL [Le01] and Microsoft's XLANG [Th01]. As the name indicates, BPEL supports the exchange of process information in web service environments. The focus lies on actually running processes rather than on general process specifications. The specification is currently under revision.

Interoperability is a topic in today's business process area. Even if BPEL processes work properly in isolation, their interaction may lead to problems like mutual waiting and thus a deadlock of the entire collaborative process. Automatic verification of interoperability correctness is crucial for guaranteeing proper behaviour of trans-organisational processes. Two problems complicate verification of collaborative processes: the merged process is usually too big to be analysed and the partners cannot be expected to publish their processes in entire detail. Thus, techniques are required to overcome these problems. In [RSS05] the notion of *controllability* has been introduced. Intuitively, a service is *controllable* if there exists a partner such that both can act and interact properly [LMSW06]. There are several approaches for providing a formal semantics for BPEL [FFK04, FGV04, St04, VA05]. Based upon these considerations, Hinz et. al. propose a translation of BPEL specifications into workflow nets [HSS05]. The purpose of this translation is less to provide a better comprehension or visualisation than to have a model which supports model checking. The transformation result, however, is only an *interpretation* since the semantics of BPEL processes is not formally defined. Assuming a workflow net semantics for BPEL, controllability can be proved automatically by constructing an automaton that describes the interaction of a workflow. This automaton contains all sufficient information while preserving inner details of the process such that correct interoperability with processes that act according to this automaton can be guaranteed. However, all these approaches are based on an interpretation of the BPEL specification only.

3 Controllability of Workflow net Interpretations of BPEL processes

The notion of *controllability* has been developed for BPEL processes on top of a workflow net [AH02] semantics for these processes [St04]. Using pattern for basic and structured elements of BPEL processes they are transformed automatically following canonical building rules [HSS05] into *open workflow nets* (oWFN) [MRS05]. oWFN are a more general kind of workflow nets having a *set* of input places and a *set* of output places instead of having exactly one each.

The *composition* [LMSW06] of two oWFNs N and M (denoted as $N \oplus M$) results in one oWFN where the joint places and initial markings of both nets are merged. Joint places are required to be input places of the one net and output places of the other, i.e. they do not share input, output and inner places, respectively. In order to define controllability we first describe what a proper (correct) behaviour of an oWFN means. Similar to the soundness criterion for workflow nets, for oWFN the criterion weak-soundness has been developed, which intuitively states that every started process must terminate, no object within the

process may remain ignored within the process after termination and there is no futile activity within the process, i.e. for every activity there is at least one run of the process that contains this activity. An oWFN is *weak-sound* [Ma03] if from every *reachable marking* a final marking is reachable and if in every final marking only output places are marked. Having defined composition and weak-soundness, a given oWFN N is *controllable* if there exists an oWFN S such that $N \oplus S$ is weak-sound. Then, S is also called a *strategy* of N.

For a given oWFN N, a strategy S can be generated automatically by constructing an *interaction graph* that represents the behaviour of S [We04]. The interaction graph is a directed graph where the arcs represent the incoming and outgoing messages from and to N and each node of the graph represents the state(s) of N until N sends or receives a new message. Any newly sent or received message leads to another node of the interaction graph. If all terminal nodes of the interaction graph contain only terminal states of N corresponding to the definition of weak-soundness, then N is controllable. In [MS05] it has been shown how interaction graphs can be extended in order to represent not only some but all strategies of N by using the concept of *operating guidelines*.

The disadvantage of this approach is that it makes use of the reachability set and not of the structure of the synthesised workflow net. In opposite to this, the method introduced next uses the original EPC models as input only.

4   EPC Model Integration

Any specification of communication between EPC models requires coupling the participating models in some form. A tight coupling by merging equally interpreted elements is reported in [SM06, MS06]. Also the origins of the presented work - like for the discussed controllability/usability - lie in the coupling of a kind of workflow nets called *module nets* which might be explained here first to provide a better understanding of Section 5.

Formal integration of processes has been reported for Petri nets in general in [BDK01] and for Process Algebra in [BPS01, Fo00b]. The absence of a proper visualisation and some restrictions concerning the semantics prohibits using Process Algebra for the modelling of businesses [Aa05]. These limitations are solved by the Semantic Process Language (SPL) which provides means for a formal specification of process sets that can be verified against non-trivial process-like properties. Since the semantics of SPL formulas (called *modules*) is defined via their Petri net implementation, their visualisation is implicitly given [Si06]. Consequently, it is also possible to simulate the models and verify their properties with all existing Petri net analysing techniques. Moreover, the approach supports stepwise development [Si05].

The formulas of SPL are built over elementary processes which specify the *occurrence* or *prohibition* of an action. Elementary processes are linked to each other using operators for *sequence* building, *alternatives*, *concurrency* and *synchronisation*, *iteration*, and *negation*. Canonical building rules generate from modules *module nets* - Petri nets with explicit *start* and *goal* transitions. Each firing sequence of such a net which reproduces the empty initial marking by firing *start* and *goal* transitions exactly once is interpreted as a *process* of the module net and, by definition, also of the original module.

124

These definitions have some important similarities with the issues relevant for the transformation of BPEL processes into EPC models. The constructive operators of BPEL correspond in many details to the operators of SPL. Moreover, proving in SPL makes use of the original input models and does not depend on the reachability set. The methods developed for SPL may therefore be applied to EPC models as well.

One central operator for proving in SPL is the *synchronisation* operator which defines the co-execution of the operands' processes such that they are merged in shared actions. This operator is implemented in module nets by *joining* all equally interpreted transitions (i.e. transitions which specify the occurrence or prohibition of the same action) and the *start* and *goal* transitions. It yields, in principle, in the intersection of the process sets of the participating modules concerning common actions.

Although the approach has been used already for the modelling, analysis and optimisation of business processes in some practical cases, it is - in opposite to EPC models - not established yet. It was therefore an aim to transfer the verification technique to EPC models which is in principle possible since in both cases only the original models and not a reachability analysis is used for verification. At the example of two EPC models taken from the SAP reference model, the formal integration of such models is demonstrated. Both processes describe how a customer inquiry about products is received, processed, and a quotation is created from the inquiry (the first one is taken from the Project Management branch of the SAP reference model, the second process EPC stems from the Sales and Distribution branch). Figure 1 shows the exemplary processes.
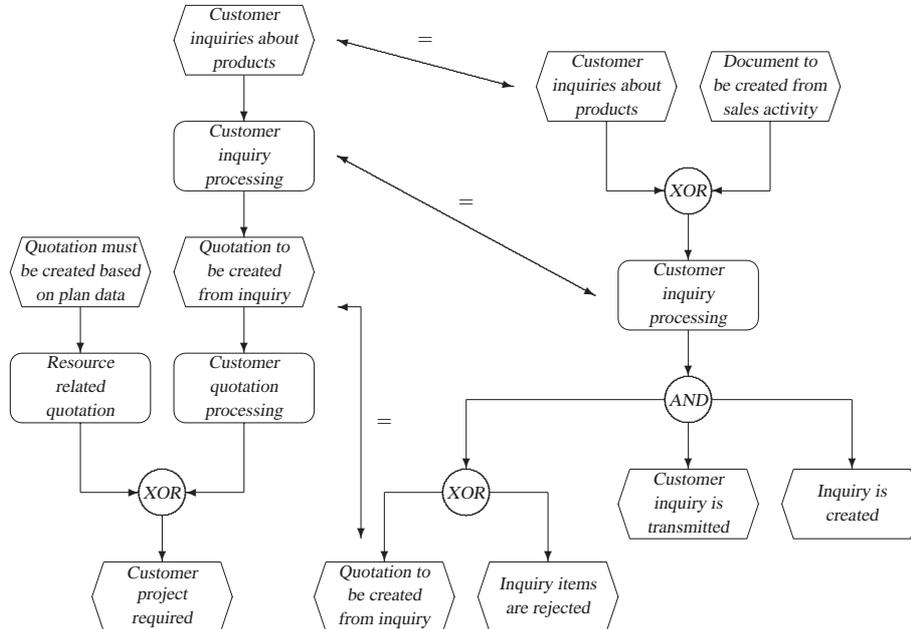

Figure 1: *Customer Inquiry* (left) and *Customer Inquiry and Quotation Processing* (right)

125

Indicated by equal names, the processes share two events and one function. Joined events are caused by all previous functions and triggered by all follower functions of the original EPC. Joined functions share their previous pre- and post-conditions (in terms of events). Figure 2 shows the result of the join. Redundant connectors are drawn with grey lines and can be omitted. The resulting EPC contains the intersection of the processes represented by the operands' EPCs.

5 Data Exchange between and Usability of EPC models

Within BPEL specifications, elementary and structured activities are distinguished. *Invoke* (sending a message) and *receive* (receiving a message) are the most important elementary activities concerned with the exchange of information between services. They are therefore in the centre of the following considerations.

A first approach to represent *invoke* and *receive* in EPC diagrams is by coupled combinations of functions and events as demonstrated in Figure 3.

In principle, information sent successfully (event *invoked*) is the start event for an asynchronous receive. However, as the implementation of the *receive* activity as an augmented workflow net in [RSS05] demonstrates, this solution does not take into account the complexity of both activities that must be operated in a workflow management system.

In a workflow management system, different instances of the same kind of process might be executed in parallel which are distinguished with the aid of a *correlation set variable*. A sent/received message must then include an identifier which enables a service to decide on whether it is addressed by the message or not. If it is not addressed, the service should return into its initial state. Before operating a received information, also a flag should be set which indicates that the message has been received properly or whether a failure has occurred. In case of a successful message transfer, also the received information should be published in an interface. Figure 4 shows the implementation of this interpretation of the processing of a *receive* as an extended EPC, i.e. besides the pure process structure also the used information objects are shown. This implementation does not include a verification of the port type which is the case in the example of the following section. Concerning the model of Figure 4 a critical remark must be made: it probably is much too close to an implementation of the *receive* activity than this is typical for an EPC model. On the other hand, it is still on a more abstract level than the respective model shown in [RSS05] which uncovers many internal details of this activity. Although this accuracy is required for a formal verification, it is obviously over-specified if the possibility to couple business processes must be discussed by modellers who want to use the service. For this purpose, the shown EPC model seems to be more appropriate.

The presented EPC model of Figure 4 explains a modeller that for successfully using/controlling the *receive* activity a proper correlation set selection is required. S/He can, moreover, recognise that the acceptance of a transmitted message is based on a further check routine which must be taken into account if the *receive* activity is used. The following section applies these observations to a practical example.

Customer inquiries about products

AND

Document to be created from sales activity

XOR

AND

Customer inquiry processing

AND

AND

XOR

Customer inquiry is transmitted

Inquiry is created

AND

Inquiry items are rejected

Quotation must be created based on plan data

Quotation to be created from inquiry

Resource related quotation

Customer quotation processing

XOR
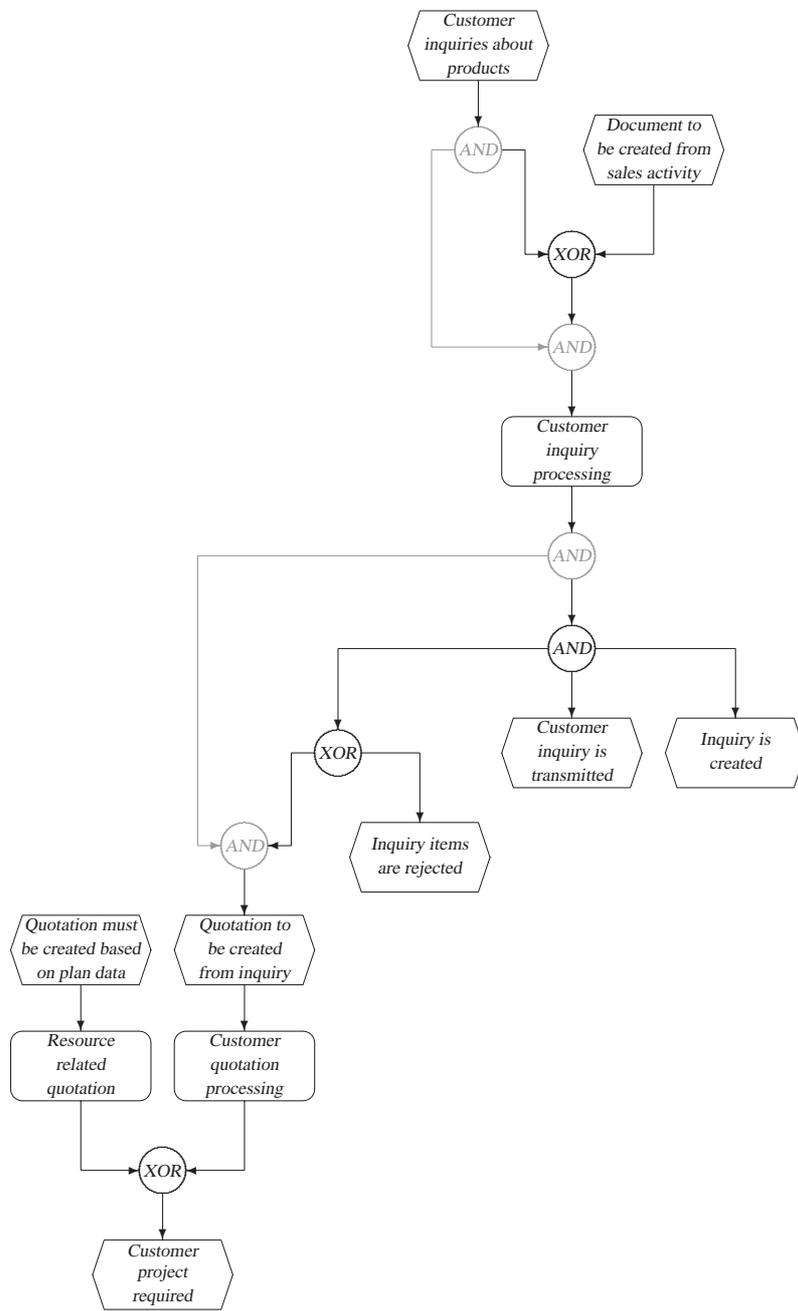
Customer project required

Figure 2: Integrated EPC of *Customer Inquiry* and *Customer Inquiry and Quotation Processing*
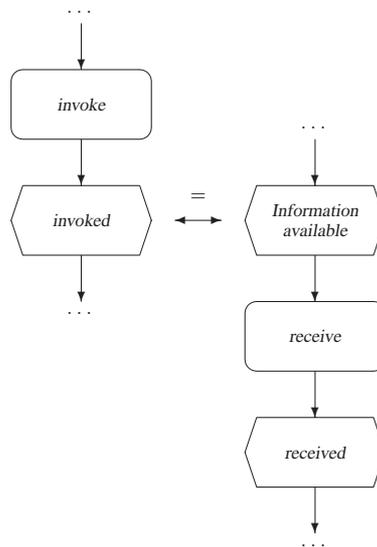
127

Figure 3: First implementation of *invoke* and *receive*

6 An E-Government Example

As a result of new political and economic agendas Public service provision in Europe has been undergoing dramatic transformation for the last couple of years. Consequently, there have been acute pressures caused by the swelling demand for both existing and new types of public services. Challenging these changes, new information and communication technologies are now opening up new opportunities to the public sector and the promotion of public services. The installation of information society applications to provide significant increase in the public sector efficiency is sub-summarised under the term of E-Government [FO00a].

On the downside, the management of the organisational change often exceeds the abilities of public institutions [SKH03] - especially the ones of smaller institutions or municipalities. On top of that, process reorganisation - necessary to offer the enhanced services in the public sector - must often have to stop short of established structures [SZ97] or legal constraints [AO05].

The introduction of web services as a solution for the establishment of E-Government could overcome those problems [SG01]. Precondition, of course, is that the technical abilities are sufficiently provided and, correspondingly, the concept of business process modelling is more and more accepted by the public authorities as well [WT03]. As a demonstration example, we analyse a web service that is currently in the evaluation period (though fully available): issuing permission on importing/exporting goods by the German federal exportation authority, the *Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA)*.
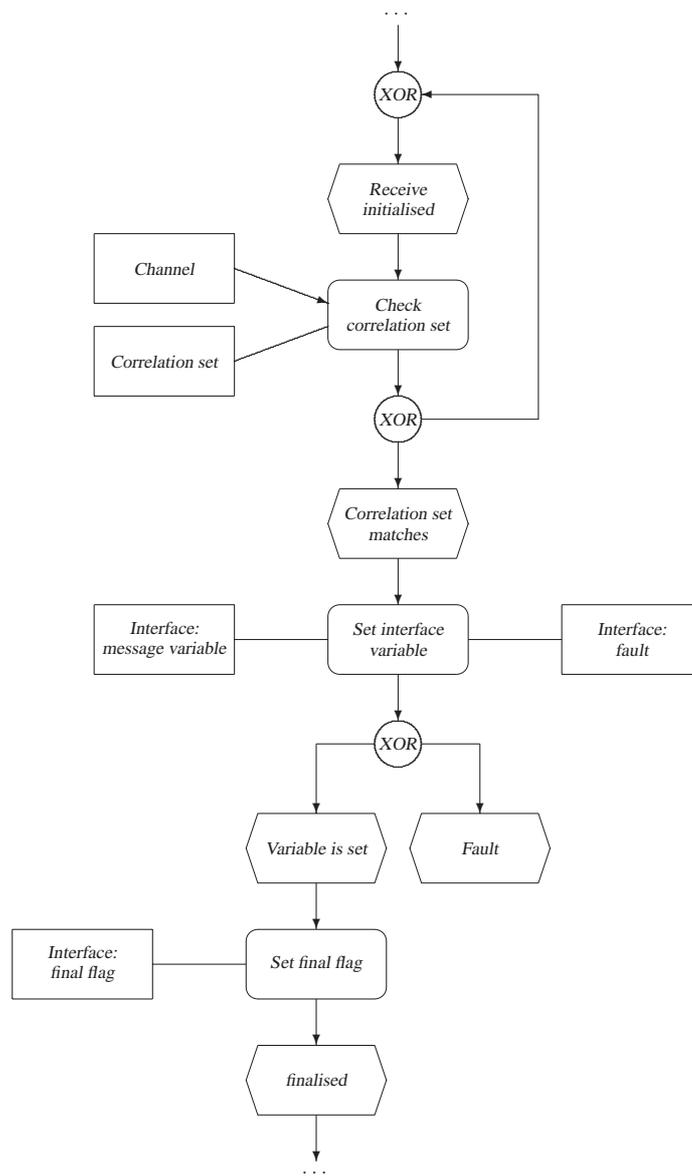
Figure 4: Implementation of *receive* specifying the internal behaviour of this activity

The BAFA is the executing authority in Germany for permissions on cross-border transactions. It takes its decisions depending on the exported/imported goods and the country the goods are exported/imported to. The BAFA does, however, not check these goods or their shipping destination/origin itself (this is duty of the customs), but rather the information on the product and whether exporting/importing of this certain item interferes with the interests of the Federal Republic of Germany or any other European Union member state. If not, an import/export certificate is issued.

To decide which products are permitted for importing/exporting from/to which country, the BAFA uses several product and country lists. The content of these lists depends on the current political debate in the European parliament and describes criteria on several categories of items that are forbidden for import/export. For example, it is not permitted to import goods to Europe that are preserved by European law (e.g. rare types of tropic wood, corals, animals) and many types of weapons or chemicals are generally forbidden for export (e.g. weapons of mass destruction). Some of these regulations might depend on a certain country or region, for example if there is an international embargo.

Though these lists are quite voluminous and change frequently, the core permission process at the BAFA is quite simple: either the applicant's information is found on a list and the import/export certificate is denied, or a specific good is free for import/export. The process is extended, if the product is more complex - for instance, a milling machine can be used to produce toys and guns as well. In case of a so called *dual-use* products, the process is extended by checking the exact purpose of the product and the background of vendor and buyer. For this, the BAFA cooperates with other institutions in various countries (i.e. customs, police, financial offices). Since the extended process bares little potential for automation, it is out of the scope of our analysis.

Already in the year 2000, the German E-Government Initiative *BundOnline 2005* uncovered the potential which lies in the automation of BAFA application processes and underlined the importance to electronically support these federal authorities' processes [BM01]. An electronic application system form (ELAN) was developed and published on the web page of the BAFA.[1] Since then, an applicant is able to register via a web interface and fill in the data. The procedure offers two major advantages: exporters do not have to fill in similar data more than once anymore and the BAFA could automate a large amount of its processes in the backoffice.

Yet, the information still needs to be typed in manually even though precise information of product and destination/origin is mostly electronically available in the applicant's database. In order to automate the overall process, the BAFA recently created a web service as an interface to its own electronically stored data which contains the regulations (lists) on goods and countries.[2] Now, the (general) information on whether permission for a specific product is needed can be answered via this web service immediately.

The class which implements the web service is called *TransmissionService*. For a user of the web service, the public methods *transferApplication()* and *submitApplication()* are of special interest. Each method requires a parameter on its call: *transferApplication()*

---

[1] http://www.bafa.de
[2] The ELAN web service can be reached at https://fg01.bafa.bund.de/elan/webservice.

```
⟨process name = ELAN - transmissionService⟩
   ...
   ⟨variables⟩
   ⟨variable name=clientData
           messageType=ClientMessage/⟩
   ...
   ⟨/variables⟩
   ⟨sequence⟩
      Client(initiate)
         ⟨receive partner=applicant
                 name=receiveData
                 PortType=receivePortType
                 operation=TransmissionService()
                 variable=ClientData/⟩

         ⟨flow⟩
            ...  internal software check
         ⟨/flow⟩
         ⟨reply partner=applicant
                name=sendmessage
                PortType=sendPortType
                operation=sendAnswer()
                variable=permission/⟩
   ⟨/sequence⟩
⟨/process⟩
```
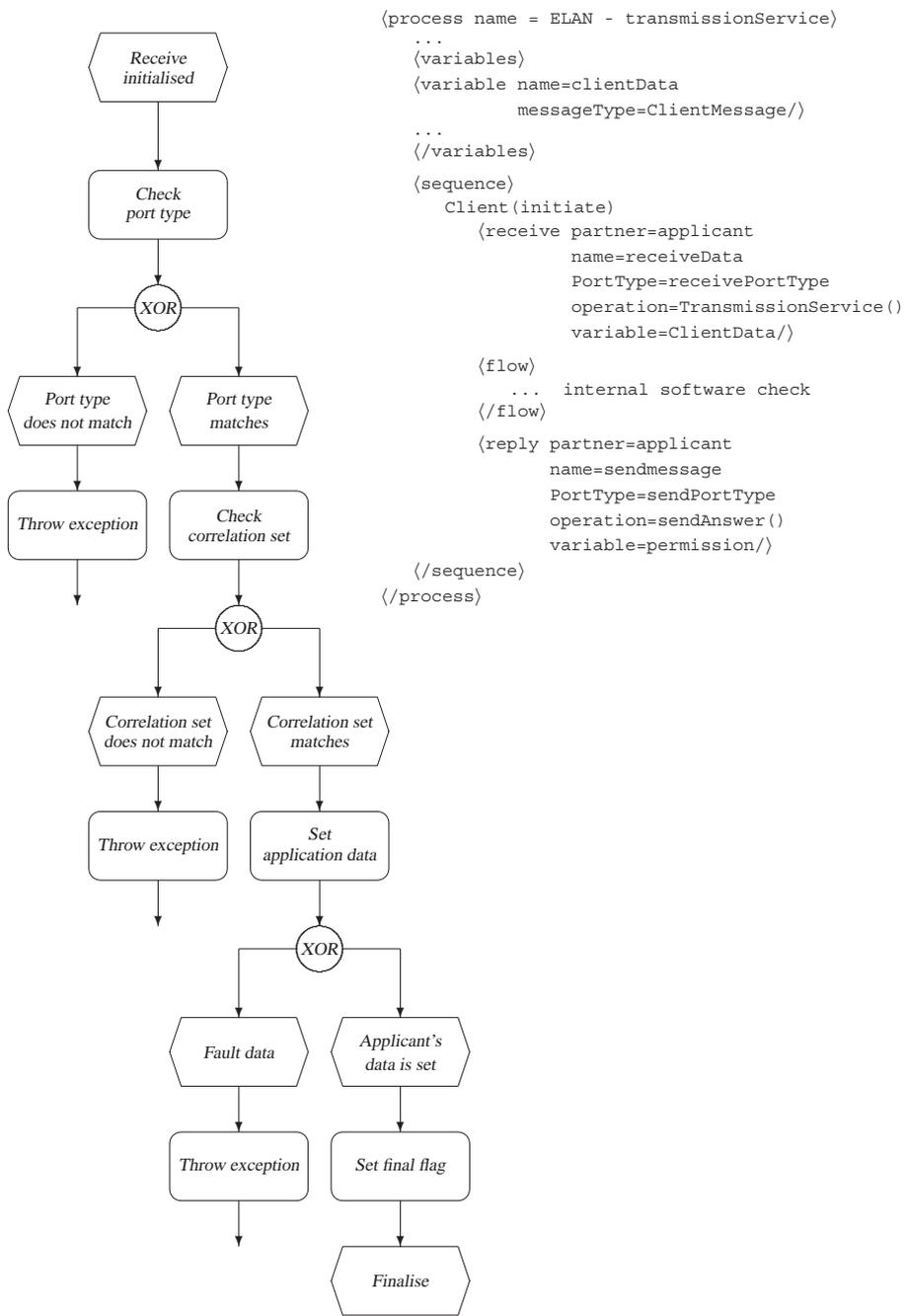
Figure 5: BPEL process of the example

131

requires a string which contains the application itself as an XML file, *submitApplication()* contains the user data. Since the correctness of the user data is checked before the web service is invoked, we focus on modelling the *TransmissionService*, which belongs to the implementation of *receive*. The *receive* activity is the one that differs from our theoretical models. As Figure 5 illustrates, we can derive both, BPEL process and EPC usability graph.

As explained in the previous paragraph, the *invoke* activity only occurs if the user is positively identified. Consequently, we can assume that, for the *receive* process to be usable, it needs to check the port type, the correlation set and the applicant's data (variables) as demonstrated in Figure 4. The EPC diagrams of Figure 5 and Figure 4 differ slightly in a way that every *XOR* operator in the BAFA's model explicitly throws an exception of its own and terminates the application. The more general defined BPEL model of *receive* however, allows multiple options for the occurrence of exceptions.

We are currently negotiating with the BAFA concerning the terms of publishing our models as part of their documentation of their web services on the web site of the BAFA.

7 Conclusion

In contrast to the concept of controllability which can be formally specified and proved (see Section 3), we have discussed a less specific approach to support a modeller in providing interfaces of interacting workflows. In this paper, for important elementary BPEL activities EPC models are presented that on the one hand show the core meaning of these activities (e.g. message passing) and on the other hand explain details that may be observed for correctly interacting workflows. In order to avoid confusion with the term controllability, we use the term usability which, intuitively, has a very similar meaning: there is a partner or environment that satisfies the input and output of a workflow such that the workflow runs and terminates correctly. Unlike the formal semantics, we have chosen a rather model-based approach to discuss BPEL constructs and their proper use for correctly interacting workflow models. As the example in Section 6 shows, the integration method presented in Section 4 can be used to compose such interacting workflows.

The amount of research projects in the sector of inter-enterprise-process-integration shows the relevance of the topic. One of these projects is the E-Justice concept of the European Union which is part of the 6th Research Framework Program funded by the action plan for eEurope 2005 by the European Commission [Co05]. Next, we plan to enhance the user-interface management for Public Services [FZ06] by our method. Thus, we hope to move closer to unified and transparent process sets over the European Union.

References

[Aa05]   Aalst, W. M. P., van der: Pi calculus versus Petri nets: Let us eat "humble pie" rather than further inflate the "Pi hype". *BPTrends*. 3(5):1–11. 2005.

[ACD⁺03]   Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., D. Smith, D., Thatte, S., Trickovic, I., und Weerawarana, S. Business Process Execution Language for Web Services - Version 1.1. `ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf` (last accessed 5.9.2005). May 2003.

[AH02]   Aalst, W. M. P., van der und Hee, K., van: *Workflow Management - Models, Methods, and Systems*. MIT Press. Cambridge, MA. 2002.

[AO05]   Alpar, P. und Olbrich, S.: *Legal Requirements and Modelling of Processes in e-Government*. Electronic Journal of e-Government Volume 3 Issue 3, S.107-116. 2005.

[BDK01]   Best, E., Devillers, R., und Koutny, M.: Petri net Algebra. In: Brauer, W., Rozenberg, G., und Salomaa, A. (Hrsg.), *EATCS*. Monographs in Theoretical Computer Science. Berlin. 2001. Springer.

[BM01]   BMI: *Umsetzungsplan für die eGovernment-Initiative Bund Online 2005*. Bundesministerium des Inneren, Stabsstelle Moderner Staat - Moderne Verwaltung. Kabinettsbeschluss vom 14. November 2001.

[BPS01]   Bergstra, J. A., Ponse, A., und Smolka, S. A. (Hrsg.): *Handbook of Process Algebra*. Elsevier. Amsterdam. 2001.

[Co05]   Commission, E.: Europe 2005 - an information society for all. *http://europa.eu.int/information_society/eeurope/2005/index_en.htm (06-07-06)*. 2005.

[Da93]   Davenport, T. H.: *Process Innovation: re-engineering Work through Information Technology*. Harvard Business School Press. Boston, MA. 1993.

[FFK04]   Fisteus, J. A., Fernández, L. S., und Kloss, C. D.: Formal Verification of BPEL4WS Business Collaborations. In: Bauknecht, K., Bichler, M., und Pröll, B. (Hrsg.), *E-Commerce and Web Technologies (EC-Web '04)*. volume 3182 of *Lecture Notes in Computer Science (LNCS)*. S. 76–85. Zaragoza, Spain. 2004. Springer.

[FGV04]   Farahbod, R., Glässer, U., und Vajihollahi, M.: Specification and Validation of the Business Process Execution Language for Web Services. In: Zimmermann, W. und Thalheim, B. (Hrsg.), *Abstract State Machines 2004. Advances in Theory and Practice*. volume 3052 of *Lecture Notes in Computer Science (LNCS)*. S. 78–94. Lutherstadt Wittenberg, Germany. 2004. Springer.

[FO00a]   FOEV: *Speyrer Definition von Electronic Government*. Forschungsinstitut für öffentliche Verwaltung bei der deutschen Hochschule für Verwaltungswissenschaften Speyer, http://foev.dhv-speyer.de (2003-01-14). 2000.

[Fo00b]   Fokkink, W.: *Introduction to Process Algebra*. Springer. Berlin. 2000.

[FZ06]   Freiheit, J. und Zangl, A.: Model-based user-interface management for public services. *In: D. Remenyi (editor): Conference proceedings of the 6th European Conference on Electronic Government (ECEG), Reading, GB S. 141-151*. 2006.

[Ga83]   Gaitanides, M.: *Prozeßorganisation*. Verlag Vahlen. München. 1983.

[Ga03]   Gadatsch, A.: *Grundkurs Geschäftsprozess-Management*. Vieweg. Wiesbaden. 3.
           2003.

[HC93]   Hammer, M. und Champy, J.: *Reengineering the Cooperation*. Harper Business. New
           York. 1993.

[Ho95]   Hollingsworth, D.: *Workflow Management Coalition: The Workflow Reference Model*.
           Workflow Management Coalition. Hampshire, UK. Issue 1.1. 1995. `http://www.`
           `wfmc.org/standards/docs/tc003v11.pdf` (last accessed 5.10.2005).

[Ho04]   Hollingsworth, D.: The Workflow Reference Model 10 Years On. In: Fischer, L.
           (Hrsg.), *Workflow Handbook 2004*. Lighthouse Point, FL. 2004. Future Strategies Inc.

[HSS05]   Hinz, S., Schmidt, K., und Stahl, C.: Transforming BPEL to Petri Nets. In: Aalst,
           W. M. P., van der, Benatallah, B., Casati, F., und Curbera, F. (Hrsg.), *Business Process
           Management (BPM 2005)*. volume 3649 of *Lecture Notes in Computer Science (LNCS)*.
           S. 220–235. Nancy, France. 2005. Springer.

[Ja96]   Jablonski, S.: Anforderungen an die Modellierung von Workflows. In: Österle, H. und
           Vogler, P. (Hrsg.), *Praxis des Workflow-Managements*. S. 65–81. Braunschweig. 1996.
           Vieweg Verlag.

[JB96]   Jablonski, S. und Bussler, C.: *Workflow Management - Modeling Concepts, Architec-
           ture and Implementation*. International Thomson Computer Press. London. 1996.

[Ju04]   Juric, M. B.: *Business Process Execution Language*. Packt Publishing. Birmingham,
           UK. 2004.

[Le01]   Leymann, F. Web Service Flow Language (1.0). `http://ibm.com/webservices/`
           `pdf/WSFL.pdf` (last accessed 17.10.2005). 2001.

[LMSW06]   Lohmann, N., Massuthe, P., Stahl, C., und Weinberg, D.: Analyzing Interacting BPEL
           Processes. In: *Business Process Management, 4th International Conference, BPM
           2006, Vienna, Austria, September 5-7, 2006, Proceedings*. volume 4102 of *Lecture
           Notes in Computer Science*. S. 17–32. Springer-Verlag. sep 2006.

[Ma03]   Martens, A.: *Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Hilfe
           von Web Services*. Dissertation. Humboldt-Universität zu Berlin, Mathematisch-
           Naturwissenschaftliche Fakultät II. 2003. erschienen in WiKi: Stuttgart, Berlin &
           Paris.

[MRS05]   Massuthe, P., Reisig, W., und Schmidt, K.: An Operating Guideline Approach to the
           SOA. *Annals of Mathematics, Computing & Teleinformatics*. 1(3):35–43. 2005.

[MS05]   Massuthe, P. und Schmidt, K.: Operating Guidelines - an Automata-Theoretic Foun-
           dation for the Service-Oriented Architecture. In: Cai, K.-Y., Ohnishi, A., und Lau, M.
           (Hrsg.), *Proceedings of the Fifth International Conference on Quality Software (QSIC
           2005)*. S. 452–457. Melbourne, Australia. September 2005. IEEE Computer Society.

[MS06]   Mendling, J. und Simon, C.: Business Process Design by View Integration. In: Eder, J.
           (Hrsg.), *Proceedings: 2nd Workshop on Business Processes Design (BPD'06)*. volume
           4103 of *Lecture Notes in Computer Science (LNCS)*. S. 55–64. Wien, September, 5 -
           7. 2006. Springer.

[Po04]   Porter, M. E.: *Competitive Advantage*. Free Press. New York. 2004.

[RSS05]   Reisig, W., Schmidt, K., und Stahl, C.: Kommunizierende Workflow-Services model-
           lieren und analysieren. *Informatik Forschung und Entwicklung*. 20:90–101. 2005.

[SG01]   Spahn, D. und Gisler, M.: *eGovernment: Eine Standortbestimmung*. Haupt-Verlag,
          Bern - Stuttgart, Wien, 2. Auflage. 2001.

[Si05]   Simon, C.: Incremental Development of Business Process Models. In: *EMISA 2005,
          Development Methods for Information Systems and their Application*. Number P-75 in
          Lecture Notes in Informatics (LNI). S. 222–235. Klagenfurt. 2005. GI.

[Si06]   Simon, C.: Integration of Planning and Production Processes. In: *Mathmod 2006,
          Special Session: Petrinets: Current Research Topics and their Application in Traffic
          Safety and Automation Engineering*. Wien, Austria. 2006.

[SKH03]   Scheer, A.-W., Kruppke, H., und Heib, R.: *E-Government - Prozessoptimierung in der
          öffentlichen Verwaltung*. Springer Verlag, Berlin Heidelberg, S.5. 2003.

[SM06]   Simon, C. und Mendling, J.: Verification of Forbidden Behavior in EPCs. In: Mayr,
          H. C. und Breu, R. (Hrsg.), *Proceedings: Modellierung 2006*. Number P-82 in Lecture
          Notes in Informatics (LNI). S. 233–242. Innsbruck, Austria, März, 22.-24. 2006. GI.

[SS04]   Schmelzer, H. J. und Sesselmann, W.: *Geschäftsprozessmanagement in der Praxis*.
          Hanser. München. 4. 2004.

[St01]   Staud, J. L.: *Geschäftsprozessanalyse*. Springer. Berlin. 2. 2001.

[St04]   Stahl, C.: A Petri Net Semantics for BPEL. Technical Report 188. Humboldt-
          Universität. Berlin. 2004.

[SZ97]   Snellen, I. und Zuurmond, A.: *From Bureacrycy to Infocracy: Management through
          Information Architecture*. In: Tyler, Snellen, Zuurmond (Hrsg.), Beyond BPR in Public
          Administration, Institutional Transformation in an Information Age, Amsterdam IOS
          Press, S. 205-224. 1997.

[Th01]   Thatte, S. XLANG - Web Services for Business Process Design - Initial Public
          Draft. `http://www.gotdotnet.com/team/xml_wsspecs/xlang-c` (last accessed
          17.10.2005). 2001.

[VA05]   Verbeek, H. M. W. und Aalst, W. M. P., van der: Analyzing BPEL Processes using
          Petri Nets. In: Marinescu, D. (Hrsg.), *Proceedings of the Second International Work-
          shop on Applications of Petri Nets to Coordination, Workflow and Business Process
          Management*. S. 59–78. Florida International University, Miami, Florida. 2005.

[We04]   Weinberg, D.: Analyse der Bedienbarkeit. Diplomarbeit. Humboldt-Universität zu
          Berlin. October 2004.

[WT03]   Wimmer, M. und Traunmüller, R.: *Geschäftsprozessmodellierung in E-Government:
          eine Zwischenbilanz*. eGov days 2003 Arbeitskreis Organisation, http://falcon.ifs.uni-
          linz.ac.at/eGovProzessmodellierung/wimmer-traunmueller.pdf (20-10-2004). 2003.