Faulty EPCs in the SAP Reference Model

J. Mendling[1], M. Moser[1], G. Neumann[1], H.M.W. Verbeek[2], B.F. van Dongen[2],
and W.M.P. van der Aalst[2]

[1] Vienna University of Economics and Business Administration
Augasse 2-6, 1090 Vienna, Austria
{jan.mendling, h9950347, neumann}@wu-wien.ac.at
[2] Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{h.m.w.verbeek, b.f.v.dongen, w.m.p.v.d.aalst}@tm.tue.nl

Abstract. Little is known about error probability in enterprise models as they are usually kept private. The *SAP reference model* is a publically available model that contains more than 600 non-trivial process models expressed in terms of *Event-driven Process Chains* (EPCs). We have automatically translated these EPCs into YAWL models and analyzed these models using WofYAWL, a verification tool based on Petri nets, in order to acquire knowledge about errors in large enterprise models. We discovered that *at least 34 of these EPCs contain errors* (i.e., at least 5.6% is flawed) and analyzed which parts of the SAP reference model contain most errors. This systematic analysis of the SAP reference model illustrates the need for verification tools such as WofYAWL.

1 Introduction

There has been extensive work on formal foundations of conceptual modeling and respective languages. However, little quantitative research has been reported on the actual use of conceptual modeling [3]. Moreover, literature typically discusses and analyses languages rather than evaluating enterprise models at a larger scale (i.e., beyond "toy examples"). A fundamental problem in this context is that large enterprise models are in general not accessible for research as they represent valuable company knowledge that enterprises do not want to reveal. One case of a model that is, at least partially, publicly available is the SAP reference model. It has been described in [2,8] and is referred to in many research papers. The extensive database of this reference model contains almost 10,000 sub-models, most of them EPC business process models [7]. Fig. 1 shows the EPC model for "Certificate Creation" as an example of one of these models. The SAP reference model was meant to be used as a documentation for SAP's ERP system. It reflects Version 4.6 of SAP R/3 which was marketed in 2000. Building on recently developed techniques to verify the formal correctness of EPC models [12], we aim to acquire knowledge about how many formal modeling errors can be expected in a large repository of process models in practice, assuming that the SAP reference model can be regarded as a representative example. We will map all non-trivial EPCs in the SAP reference model onto YAWL

S. Dustdar, J.L. Fiadeiro, and A. Sheth (Eds.): BPM 2006, LNCS 4102, pp. 451–457, 2006.

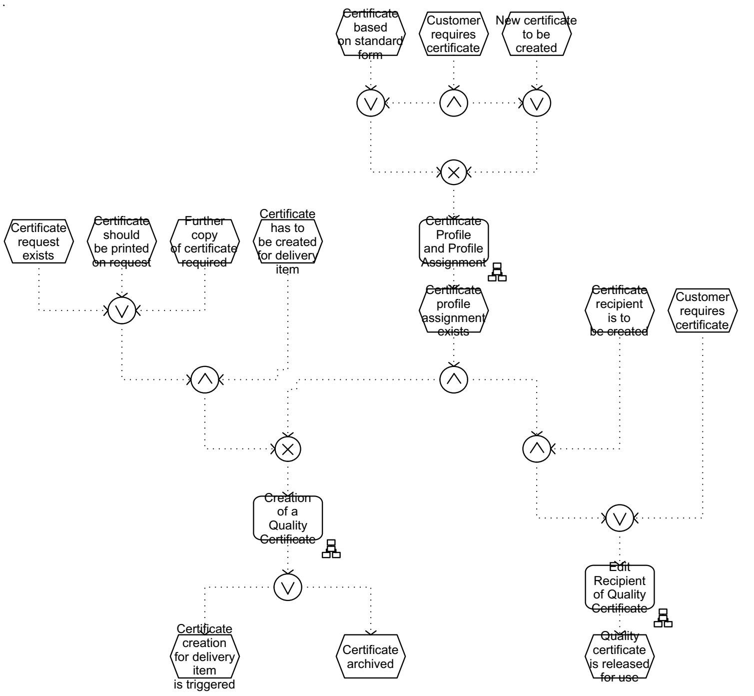

Fig. 1. One of the EPCs in the SAP reference model: the "Certificate Creation" process

models [1] and use the WofYAWL tool [12] for the verification (based on the relaxed-soundness criterion [4]). We have to stress that this analysis yields a lower bound for errors since some errors may not be discovered by this tool. Therefore, it has to be expected that there are more errors than those that we actually identify.

The remainder of this paper is organized as follows. Section 2 describes the research design. In particular, we discuss the mapping of EPCs from the SAP reference model to YAWL models, the analysis techniques employed by WofYAWL, and descriptive statistics that provide a comprehensive inventory of errors in the SAP reference model. Finally, Section 3 presents related work before Section 4 concludes with a summary of our contribution and its limitations.

2 Research Design and Results

In this section, we present the way we evaluated the SAP reference model. We use the ARIS XML export of the reference model as input to several transformation and analysis steps. In a first step, the EPC to YAWL transformation program generates a YAWL XML file for each EPC in the reference model (see Sect. 2.1). These YAWL models are then analyzed with WofYAWL that produces an XML error report of design flaws (see Sect. 2.2). Furthermore, we extract

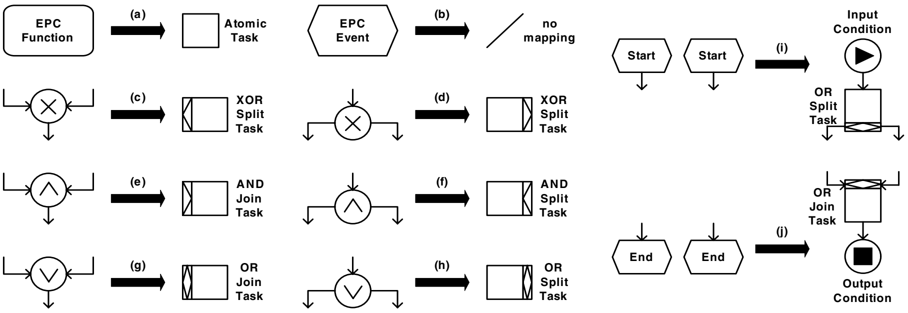

Fig. 2. Overview of the EPC to YAWL Mapping

descriptive information such as the number of elements of a certain element type and whether there are cycles for each EPC model (see Sect. 2.3).

2.1 Transformations of EPCs to YAWL

Several mappings from EPCs to Petri Nets have been proposed in order to verify formal properties, see e.g. [9] for an overview. In this paper, we use a transformation from EPCs to YAWL as defined in [10]. The advantage is that each EPC element can be directly mapped to a respective YAWL element (see Fig. 2). Even though EPCs and YAWL are very similar in this sense, there are three differences that have to be considered in the transformation: state representation, connector chains, and multiple start and end events.

EPC functions can be mapped to YAWL tasks following mapping rule (a) of Fig. 2. The first difference between EPCs and YAWL is related to *state representation*. EPC events define pre- and post-conditions of functions. They do not capture state directly. Therefore, rule (b) defines that events are not mapped to YAWL taking advantage of the fact that arcs in YAWL represent implicit conditions if they connect two tasks. In EPCs connectors are independent elements. Therefore, it is allowed to build so-called *connector chains*, i.e. paths of two or more consecutive connectors (cf. Fig. 1). In YAWL there are no connector chains since splits and joins are part of tasks. The mapping rules (c) to (h) map every connector to a dummy task with the matching join or split condition (see Fig. 2). The third difference stems from *multiple start and end events*. An EPC is allowed to have more than one start event and more than one end event. In YAWL there must be exactly one start condition and one end condition. Therefore, the mapping rules (i) and (j) generate an OR split for multiple starts and an OR join for multiple ends. Fig. 3 gives the result of applying the transformation to the “Certificate Creation” EPC of Section 1. Note that connectors are mapped onto dummy tasks. To identify these tasks they are given a unique label extracted from the internal representation of the EPC, e.g., task “and (c8z0)” corresponds to the AND-split connector following event “Customer requires certificate.”

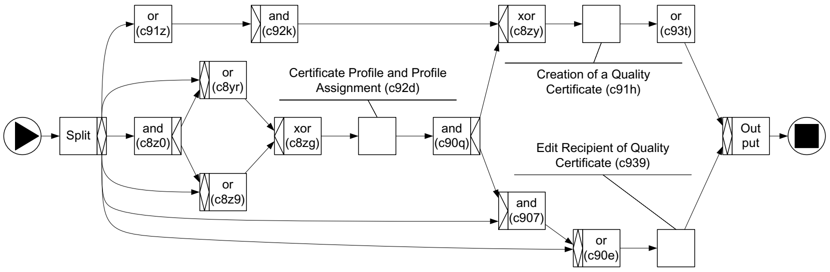

Fig. 3. YAWL model obtained by applying the mapping shown in Fig. 2 to the example

2.2 WofYAWL Analysis

After mapping the EPC onto YAWL, we can use our verification tool WofYAWL [13]. WofYAWL first maps a YAWL model onto a Petri net [11]. Fig. 4 sketches a small fragment of the Petri net that results from mapping the YAWL model of Fig. 3. The fragment only considers the dummy tasks resulting from the mapping of the top four connectors in Fig. 1. Moreover, from the initial OR-split task "Split" we only consider the arcs connected to these four dummy tasks.

The "happy smileys" in Fig. 4 are used to identify net elements that are involved in so-called "good execution paths", that is, the execution paths in the Petri net that lead from the initial state to the *desired* final state. In Fig. 4, there exist two such paths, which join at the XOR-join named "xor (c8z9)". The "sad smileys" visualize relevant parts in the Petri net that are not covered by some good execution path. WofYAWL issues respective warnings. These indicate a problem involving the top four connectors in Fig. 1. Note that the AND-split connector splits the flow into two paths that join with an XOR-join. Hence these two paths cannot be involved in a good execution. Moreover, if the AND-split connector is not allowed to occur, the two OR-joins could as well be XOR-joins. In our analysis we use *transition invariants* to avoid constructing large or even infinite state spaces [12]. Moreover, we have used existing Petri-net-based reduction rules [11] to further reduce the complexity of the models without loosing any information. For further details on this approach we refer to [12].

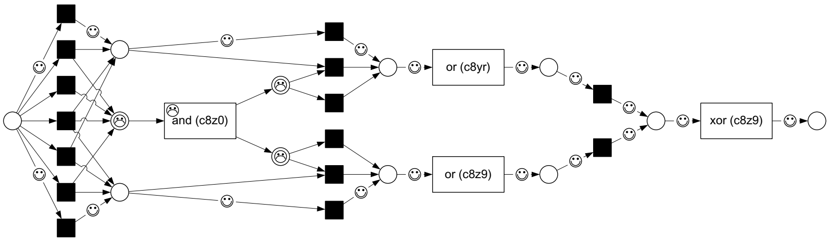

Fig. 4. Petri net fragment of the converted YAWL model

Table 1. Branches of the SAP Reference Model. The columns $E_{av.}$, $F_{av.}$, $C_{av.}$, $A_{av.}$ refer to the mean number of events, functions, connectors, and arcs.

Branch	Model	%	EPC	%	$E_{av.}$	$F_{av.}$	$C_{av.}$	$A_{av.}$	Cycle	Error	%
Asset Accounting	461	4.7%	43	7.1%	13.9	4.0	5.2	23.3	0	7	16.3%
Benefits Administration	50	0.5%	6	1.0%	9.5	3.3	5.8	19.7	3	0	0.0%
Compensation Management	122	1.2%	18	3.0%	7.6	3.4	3.3	13.7	3	1	5.6%
Customer Service	402	4.1%	41	6.8%	16.5	3.6	9.0	29.5	3	1	2.4%
Enterprise Controlling	599	6.1%	22	3.6%	14.3	10.1	6.1	32.1	0	3	13.6%
Environment, Health, Safety	102	1.0%	19	3.1%	3.5	2.7	1.2	7.0	0	0	0.0%
Financial Accounting	614	6.2%	54	8.9%	13.0	4.0	5.1	21.8	0	3	5.6%
Position Management	4	0.0%	0	0.0%	0.0	0.0	0.0	0.0	0	0	n.a.
Inventory Management	184	1.9%	3	0.5%	15.0	7.0	6.0	28.0	2	0	0.0%
Organizational Management	37	0.4%	5	0.8%	12.0	3.0	6.6	24.0	3	0	0.0%
Payroll	541	5.5%	7	1.2%	5.7	3.1	2.1	11.4	0	1	14.3%
Personnel Administration	15	0.2%	4	0.7%	7.3	1.5	4.0	12.3	0	0	0.0%
Personnel Development	60	0.6%	10	1.7%	8.7	2.5	4.4	15.6	3	1	10.0%
Personnel Time Management	87	0.9%	12	2.0%	10.8	3.0	5.3	19.5	1	2	16.7%
Plant Maintenance	399	4.1%	35	5.8%	20.5	4.2	11.4	37.8	9	1	2.9%
Procurement	444	4.5%	37	6.1%	6.7	3.5	2.7	12.4	0	2	5.4%
Product Data Management	366	3.7%	26	4.3%	4.5	5.4	2.2	13.7	0	0	0.0%
Production	296	3.0%	17	2.8%	8.8	3.0	2.9	13.7	0	1	5.9%
Production Planning	194	2.0%	17	2.8%	5.7	2.9	3.0	11.5	0	0	0.0%
Project Management	347	3.5%	36	6.0%	8.5	3.8	2.2	14.0	0	0	0.0%
Quality Management	209	2.1%	20	3.3%	20.5	3.8	11.7	37.8	1	1	5.0%
Real Estate Management	169	1.7%	6	1.0%	12.7	6.5	7.3	27.0	1	1	16.7%
Recruitment	56	0.6%	9	1.5%	7.4	2.6	4.1	13.8	3	0	0.0%
Retail	842	8.6%	1	0.2%	7.0	5.0	2.0	11.0	0	0	0.0%
Revenue & Cost Controlling	568	5.8%	19	3.1%	16.5	10.2	7.9	36.0	1	1	5.3%
Sales & Distribution	703	7.1%	76	12.6%	10.6	3.1	4.3	16.6	0	1	1.3%
Training & Event Management	95	1.0%	12	2.0%	13.0	2.7	6.2	22.2	0	1	8.3%
Travel Management	116	1.2%	1	0.2%	24.0	7.0	16.0	48.0	0	0	0.0%
Treasury	1761	17.9%	48	7.9%	10.5	3.5	4.5	18.1	0	6	12.5%
All 29 Branches	9844	100%	604	100%	11.5	4.0	5.2	20.8	33	34	5.6%

2.3 Descriptive Statistics

The sample of the SAP reference model that was available for this research contains 9844 models, but only a fraction of them represent proper EPCs with at least one start event and one function. There are 604 of such process models as listed in the column *EPC*. Using the transformations and the WofYAWL tool described in Sect. 2, we discovered that at least 34 models have errors (5.6% of 604 analyzed EPCs). Table 1 summarizes the SAP reference model subdivided into its 29 branches. It can be seen that the number of EPC models varies substantially (from none in Position Management to 76 in Sales & Distribution). Furthermore, the EPCs are of different size indicated by the mean number of events, functions, connectors, and arcs in columns $E_{av.}$, $F_{av.}$, $C_{av.}$, $A_{av.}$ respectively. The column *Cycle* states how many EPCs contain cycles, and *Error* for how many models WofYAWL reports an error.

3 Related Research

Work on the verification of process models can roughly be put into three categories: *verification of formal models*, i.e. the model with formal executable semantics is correct or not; *verification of informal models*, i.e. defining subclasses of informal models that are mapped onto formal models, the model is correct or not; and *verification by design*, i.e. the modeling language does not allow for syntactical errors. Examples are block structured models. For related work for each category we refer to [5]. Besides these categories, there are some verification approaches that are a combination of others. For example [6] involves the process designer in the verification process. Therefore, this approach is not applicable for the automatic verification of the entire SAP reference model. The approach we use based on WofYAWL has been introduced in [13]. Yet, it is not complete as there may be errors left undetected. Still, this paper uniquely combines formal error detection with a large set of real-world process models. This way, we have identified a lower bound of 5.6% for errors in the SAP reference model.

4 Contributions and Limitations

This paper provides a lower bound of 5.6% for the number of faulty EPCs in the SAP reference model. Our automatic verification approach is based on a mapping from EPCs to YAWL and on the utilization of the WofYAWL tool. As far as we know, this is the first systematic analysis of the EPCs in the SAP reference model. Yet, our approach still has some limitations: WofYAWL does not find all errors and the SAP reference model is only one specific case of an enterprise model. Therefore, we aim to improve the automatic detection of errors. Moreover, a analysis of further large enterprise models is needed to better understand why and when modelers introduce errors.

References

1. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
2. T. Curran and G. Keller A. Ladd. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model.* Enterprise Resource Planning Series. Prentice Hall PTR, Upper Saddle River, 1997.
3. Islay Davies, Peter Green, Michael Rosemann, Marta Indulska, and Stan Gallo. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, In Press, 2006.
4. J. Dehnert and P. Rittgen. Relaxed Soundness of Business Processes. In K.R. Dittrich, A. Geppert, and M.C. Norrie, editors, *Proceedings of CAiSE 2001*, volume 2068 of *LNCS*, pages 157–170. Springer-Verlag, Berlin, 2001.
5. B.F. van Dongen and M.H. Jansen-Vullers. EPC Verification in the ARIS for MySAP reference model database. BETA Working Paper WP 142, Eindhoven University of Technology, 2005.
6. B.F. van Dongen, H.M.W. Verbeek, and W.M.P. van der Aalst. Verification of EPCs: Using reduction rules and Petri nets. In *Conference on Advanced Information Systems Engineering (CAiSE 2005)*, volume 3520 of *LNCS*, pages 372–386. Springer-Verlag, Berlin, 2005.
7. G. Keller, M. Nüttgens, and A.W. Scheer. Semantische Processmodellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89 (in German), Saarbrücken, 1992.

8. G. Keller and T. Teufel. *SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping.* Addison-Wesley, 1998.
9. E. Kindler. On the Semantics of EPCs: Resolving the Vicious Circle. *Data and Knowledge Engineering*, 56(1):23–40, 2006.
10. J. Mendling, J. Recker, M. Rosemann, and W.M.P. van der Aalst. Generating Correct EPCs from Configured C-EPCs. In *Proceedings of the 21th Annual ACM Symposium on Applied Computing (SAC 2006).* ACM Press, 2006.
11. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
12. H.M.W. Verbeek and W.M.P. van der Aalst. On the verification of EPCs using T-invariants. BPM Center Report BPM-06-05, BPMcenter.org, 2006.
13. H.M.W. Verbeek, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Verifying workflows with cancellation regions and OR-joins: An approach based on invariants. BETA Working Paper WP 156, Eindhoven University of Technology, 2006.