Information and Software Technology 52 (2010) 127–136

Contents lists available at ScienceDirect

Information and Software Technology

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

Seven process modeling guidelines (7PMG)

J. Mendling [a,*], H.A. Reijers [b], W.M.P. van der Aalst [b]

[a] Humboldt University, Unter den Linden 6, 10099 Berlin, Germany
[b] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 21 December 2008
Received in revised form 29 July 2009
Accepted 17 August 2009
Available online 23 August 2009

Keywords:
Business process modeling
Model quality
Guidelines

ABSTRACT

Business process modeling is heavily applied in practice, but important quality issues have not been addressed thoroughly by research. A notorious problem is the low level of modeling competence that many casual modelers in process documentation projects have. Existing approaches towards model quality might be of benefit, but they suffer from at least one of the following problems. On the one hand, frameworks like SEQUAL and the Guidelines of Modeling are too abstract to be applicable for novices and non-experts in practice. On the other hand, there are collections of pragmatic hints that lack a sound research foundation. In this paper, we analyze existing research on relationships between model structure on the one hand and error probability and understanding on the other hand. As a synthesis we propose a set of seven process modeling guidelines (7PMG). Each of these guidelines builds on strong empirical insights, yet they are formulated to be intuitive to practitioners. Furthermore, we analyze how the guidelines are prioritized by industry experts. In this regard, the seven guidelines have the potential to serve as an important tool of knowledge transfer from academia into modeling practice.

1. Introduction

Since the 1970s and 1980s, conceptual modeling is a major research area in the IS field. The main motivation to engage in conceptual modeling is to reduce the chances on developing faulty requirements in the early phases of system development [1]. A recent empirical study has shown that *business processes* have become the central objects in many conceptual modeling efforts, e.g. to support their documentation, improvement and automated enactment [2]. This development can be explained by an increased focus of enterprises on those same business processes: they are perceived as the most relevant entities to be managed towards enhanced organizational performance [3].

Usability is an important quality issue of process documentations [4]. As understanding the process is a crucial task in any process analysis technique [5], the process model itself should be intuitive and easy to comprehend too. Process modeling tools, like ARIS and Casewise, have greatly eased the standardization, storage, and sharing of diagrams of process. Many enterprises have adopted such tools as they are perceived as much better alternatives to the use of pen and paper, or even general graphical drawing tools, e.g. Microsoft's Visio or Powerpoint. But despite the support that is provided by tools, users hardly get any support in creating process models that business professionals can easily analyze and understand. Adequate guidance is of particular importance as large projects on process documentation heavily rely on novices and non-expert modelers [6]. To appreciate the impact of a model that is difficult to assess, it should be realized that in the execution of a single project dozens, hundreds or even thousands of process models may be developed [7,8]. This clarifies why a process model that is immediately usable towards its purpose is of great economic benefit.

Even though some theoretical frameworks and guidelines are available in the area of process modeling, for instance SEQUAL or the Guidelines of Modeling [9,10], these typically require a certain level of modeling competence. They distinguish the major quality categories, but remain too abstract to be directly applicable by non-experts. In other words, such guidelines are hardly related to the concrete actions that process modelers undertake in capturing e.g. the steps and actors in a process. More practice-oriented and -inspired guidelines are available too, see e.g. [11]. The problem behind such guidelines is that hardly any empirical support is provided for them and, if available, it is anecdotic at best. From a research perspective, it can be noted that much of the existing work into process modeling does not focus on providing modeling support either. Rather, the interest is with the more formal side of process modeling, see e.g. [12,13].

This paper seeks to support the builders of business process models by providing them with a set of seven modeling guidelines, called 7PMG. This set is thought to be helpful in guiding users towards improving the quality of their models, in the sense that these are likely (1) to become comprehensible to various

* Corresponding author. Tel.: +49 30 2093 5805; fax: +49 30 2093 5741.
E-mail addresses: jan.mendling@wiwi.hu-berlin.de (J. Mendling), h.a.reijers@tue.nl (H.A. Reijers), w.m.p.v.d.aalst@tue.nl (W.M.P. van der Aalst).

0950-5849/$ - see front matter
doi:10.1016/j.infsof.2009.08.004

stakeholders and (2) to contain few syntactical errors. Each of these guidelines gives directions on how a process model can be improved and which alternative of a set of behavior-equivalent representations should be preferred. As such, the application of 7PMG will improve the efficiency of projects within enterprises that rely on the use of this particular type of conceptual models.

The novelty of the presented work is that all the guidelines of 7PMG build on sound scientific insights that have emerged over the past years into the relationship between process modeling styles on the one hand and both model understanding and error-proneness on the other. As of yet, these insights have not been synthesized into guidelines that are clear, practically applicable, and well-motivated. In this way, 7PMG not only contrasts other frameworks that have been criticized for lack of empirical foundation [14] but it also offers guidance that practitioners can apply in their business process centered initiatives straightaway. Finally, 7PMG provides a baseline for further research into process modeling to extend this set and to develop advanced tool support to facilitate modeling activities.

Against this background, the paper is organized as follows. Section 2 outlines the background of our research, namely different approaches towards process model quality. Section 3 presents the seven process modeling guidelines (7PMG) that we synthesize from prior research. Section 4 presents indications on how the guidelines should be prioritized. Section 5 contributes a discussion of the limitations and merits of these guidelines. Section 6 closes the paper with a conclusion.

2. Background

The roots of process modeling can be traced back to the early 20th century as a tool for organizational design (see [15]). It gained some attention as a subject of information systems research with the invention of office automation systems in the 1970s and 1980s (see [16,17]). The business process reengineering boom of the early 1990s contributed to a consolidation of the field and the definition of process modeling languages such as Event-driven Process Chains (EPCs) [18]. At the core of such languages is a representation of control flow between different activities, which can be extended with different perspectives such as organizational responsibilities or object flow [19–22]. There are mainly four streams of work that discuss guidelines and quality issues for such conceptual process models: top-down quality frameworks, bottom-up metrics related to quality aspects, empirical surveys related to modeling techniques, and pragmatic guidelines.

One prominent *top-down quality framework* is the SEQUAL framework [9,23]. It builds on semiotic theory and defines several quality aspects based on relationships between a model, a body of knowledge, a domain, a modeling language, and the activities of learning, taking action, and modeling. In essence, syntactic quality relates to model and modeling language; semantic quality to model, domain, and knowledge; and pragmatic quality relates to model and modeling and its ability to enable learning and action. Although the framework does not provide an operational definition of how to determine the various degrees of quality, it has been found useful for business process modeling in experiments [24]. The Guidelines of Modeling (GoM) [10] define an alternative quality framework that is inspired by general accounting principles. The guidelines include the six principles of correctness, clarity, relevance, comparability, economic efficiency, and systematic design. This framework was operationalized for EPCs and also tested in experiments [10]. Furthermore, there are authors (e.g. [14]) advocating a specification of a quality framework for conceptual modeling in compliance with the ISO 9126 standard [25] for software quality. A respective adaptation to business process modeling is reported in [26]. Although these works offer a good insight into quality issues of a model, they do not provide a straightforward method for implementation in a modeling project. A major problem in these projects is the sheer number of models (often more than thousand) and the low level of competence that casual modelers have [6]. Therefore, easy-to-follow guidelines are needed in practice.

For these reasons, several recent works have tried to approach this problem by studying *bottom-up metrics related to quality aspects* of process models. This area is still fragmented and authors have partially worked isolated from each other (see for an overview [15]). Several of these contributions are purely theoretical without empirical validation. Most authors doing experiments focus on the relationship between metrics and quality aspects: Canfora et al. study the connection mainly between count metrics – for example, the number of tasks or splits – and maintainability of software process models [27]; Cardoso validates the correlation between control flow complexity and perceived complexity [28]; and Mendling et al. use metrics to predict control flow errors such as deadlocks in process models [29,30]. The results reveal that an increase in size of a model appears to have a negative impact on quality. Further work by Mendling et al. investigate the connection between metrics and understanding [31,32]. While some metrics are confirmed regarding their impact, also personal factors of the modeler – like competence – are revealed as important for understanding.

There are some *empirical surveys* related to modeling techniques. In [33] the authors study how business process modeling languages have matured over time. While this is valuable research, it does not reveal insights on single, concrete process models. The same holds for [34] who study the usability of UML. In [35] the authors also approach understandability, not of individual process models, but on the level of the modeling language. They find out that EPCs seem to be more understandable than Petri nets. Contrarily, [36] find that model users' knowledge of the exact modeling notation is of negligible influence. The model of [37] investigates the notion of process modeling success. Several factors are identified as important (beyond modeling-related aspects) including stakeholder participation, management support, project management, information resources, and modeler's expertise.

Pragmatic guidelines have been proposed in different practitioner outlets. As it is difficult to provide an exhaustive account of such guidelines from practice, we discuss only some proposals here. In [11], 10 tips for process modeling are summarized. Several of these tips like "make your models hierarchical" and "make your models valid" do not directly provide an answer how this should be done in practice. One of the most tangible rules "label activities verb–noun" has been suggested by other practitioners before, see e.g. [38,39]. It is the only guideline that is operational and that has been analyzed empirically. In [40] it is found that it results in better models in terms of understanding than alternative labeling styles.

The lack of modeling expertise has been mentioned as a motivation for several of the above mentioned works. It is confirmed by high error rates in real-world modeling projects (between 10% and 20%) [41–43]. Clearly, there is a need for simple, yet well-founded guidelines. In the following section, we synthesize results from empirical research in this area and formulate seven process modeling guidelines.

3. 7PMG

In this section, we introduce the seven process modeling guidelines (7PMG) that we synthesize from empirical work. In Section 3.1 we describe an example process from a Dutch governmental agency that we use to illustrate the guidelines. Sec-

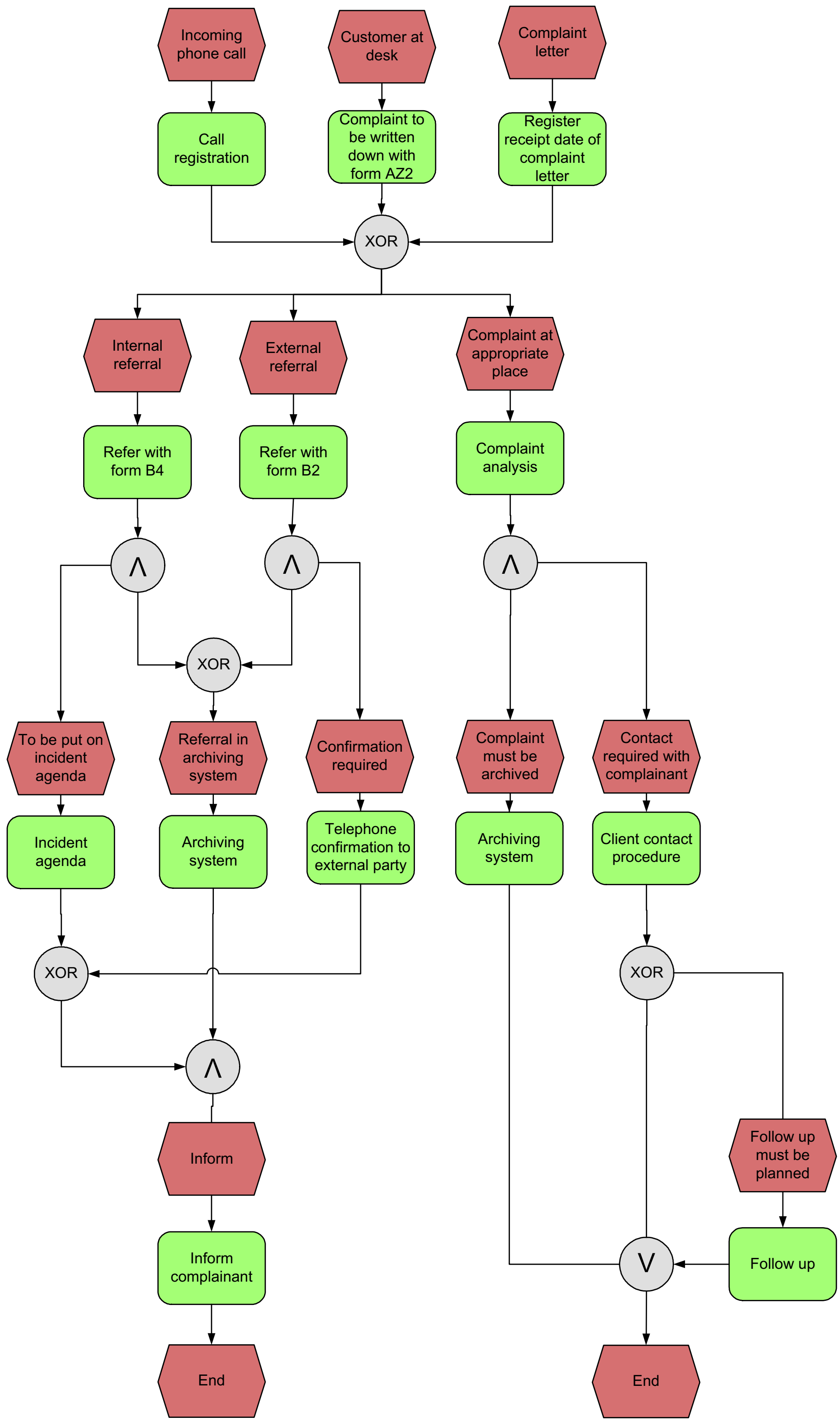

Fig. 1. The original complaint process model.

tion 3.2 summarizes empirical research upon which the seven guidelines are built. Section 3.3 presents each guideline in detail and explains the foundation. Section 3.4 shows how using the guidelines can help to identify improvements of the example model.

3.1. An example process

To illustrate 7PMG, we use the running example shown in Fig. 1. This model describes the complaint process of a Dutch governmental agency as it was modeled by the people in this organization. It was constructed without considering the guidelines and, as will be shown later, can be improved using our guidelines.

The model in Fig. 1 follows the Event-driven Process Chains (EPC) notation [18,22], one of the most popular modeling techniques in industry. In an EPC, so-called *functions* (green rectangles) correspond to the various tasks that may need to be executed (e.g. "Register receipt date of complaint letter"). *Events* (red hexagons) describe the situation before and after a function is executed (e.g. "Customer at desk"). *Logical connectors* (grey circles) define routing rules. In particular, there are three types of connectors: the logical AND for concurrency, XOR for exclusive choices, and OR for inclusive choices. Functions, events, and connectors are the classical elements of control flow modeling. These routing elements are also included in other modeling languages like BPMN, YAWL, or UML activity diagrams.

The given model roughly describes the following procedure for handling complaints. A new case is opened if a new complaint is received – be it as a phone call, as a personal contact, or as a letter. In some situations, the complaint must be referred, either internally or externally. Internal referrals have to be put on the incident agenda while external referrals require a confirmation. In both cases the referral is archived in parallel. Finally, the complainant is informed. If no referral is required, a complaint analysis is conducted. Later, the complaint is archived and the complainant is contacted, with an optional follow up.

3.2. Empirical research background

This section describes the research background upon which we define the seven process modeling guidelines. The usability of process models is strongly connected with its ease of comprehension. In prior research, we have investigated the connection between structural characteristics of a process model and different factors of comprehension including *process model understanding*, *error probability*, and *label ambiguity*. Below we describe the respective experiments.

- *Process model understanding* relates to the degree to which a process model can be easily understood. In an experiment reported in [31] we investigate the connection between process model understanding and structural properties of the model. We used a questionnaire and had it filled out by 73 students who followed courses on process modeling at the Eindhoven University of Technology, the University of Madeira, and the Vienna University of Economics and Business Administration. We found that several structural metrics showed a negative correlation with understanding, including the number of OR-joins and the average degree of connectors.
- *Error probability of process models* captures to what extent a modeler is able to still extend a process model without introducing errors. In different experiments we have determined prediction functions for error probability using the correctness notions of relaxed soundness and the 600 EPCs of the SAP Reference Model [41] and using EPC Soundness for a collection of 2000 EPCs from industry [44]. Both prediction functions trace error probability back to structural metrics of the process models. We have found that size and complexity are an important drivers of error probability.
- *Ambiguity of activity labels* is a significant road block to the understanding of a process model. We have observed that there are different grammatical style used in the SAP Reference Model, in particular the verb-object style ("send letter") and the action-noun style ("letter sending") [40]. We conducted an experiment and found that verb-object labels were considered less ambiguous and more useful by 29 post-graduate students from Eindhoven University of Technology in the Netherlands.

Based on these empirical insights into the three aspects of process model comprehension, we define the seven process modeling guidelines.

3.3. The guidelines

7PMG provides a set of recommendations on how to build a process model from scratch as well as for improving existing process models. Each of the guidelines builds on empirical research described above [31,41,44,40]. It is important to note that 7PMG builds on the insight that there are different ways to describe the same behavior using a process model. Respective notions of formal behavior equivalence like bisimulation have been heavily researched from a verification perspective [45]. 7PMG identifies desirable properties that can be used as directions when changing a process model to a behavior-equivalent, but more understandable model. The guidelines are as follows:

G1: Use as few elements in the model as possible. The size of the model has undesirable effects on understandability and likelihood of errors: Larger models tend to be more difficult to understand [31] and have a higher error probability than small models [41,44].

G2: Minimize the routing paths per element. The higher the degree of an element in the process model, i.e. the number of input and output arcs together, the harder it becomes to understand the model [31]. As shown in [44] there is a strong correlation between the number of modeling errors and the average or maximum degree of elements in a model.

G3: Use one start and one end event. The number of start and end events is positively connected with an increase in error probability [44]. Most workflow engines require a single start and end node [46]. Moreover, models satisfying this requirement are easier to understand and allow for all kinds or analysis (e.g., soundness checks).

G4: Model as structured as possible. A process model is structured if every split connector matches a respective join connector of the same type. Structured models can be seen as formulas with balanced brackets, i.e., every opening bracket has a corresponding closing bracket of the same type. Unstructured models are not only more likely to include errors [44], people also tend to understand them less easily [31].

G5: Avoid OR routing elements. Models that have only AND and XOR connectors are less error-prone [44]. Furthermore, there are some ambiguities in the semantics of the OR-join leading to paradoxes and implementation problems [47].

G6: Use verb-object activity labels. A wide exploration of labeling styles that are used in actual process models, discloses the existence of two popular styles and a rest category [48]. From these, people consider the verb-object style, like "Inform complainant", as significantly less ambiguous and more useful than action-noun labels (e.g. "Complaint analysis") or labels that follow neither of these styles (e.g. "Incident agenda") [40].

G7: Decompose the model if it has more than 50 elements. This guideline relates to **G1** that is motivated by a positive correlation between size and errors. For models with more than 50 elements the error probability tends to be higher than 50% [44]. Therefore, large models should be split up into

Table 1
Overview 7PMG.

G1	Use as few elements in the model as possible
G2	Minimize the routing paths per element
G3	Use one start and one end event
G4	Model as structured as possible
G5	Avoid OR routing elements
G6	Use verb-object activity labels
G7	Decompose a model with more than 50 elements

smaller models. Large sub-components with a single entry and a single exit [49] can be replaced by one activity that points to the original sub-component as a separate models.

These seven guidelines are summarized in Table 1.

3.4. Application

To illustrate 7PMG, we will show how the recommendations can be used to transform the original model that captures the complaint handling process. In Fig. 2, the same procedure is shown, but various areas of the model have now been marked and labelled with guideline identifiers. In Fig. 3, a transformed model is shown which results from the application of 7PMG. In what follows, we will consider the application of the guidelines one by one.

The original model has a problem of redundant information since there are lots of events that do not give additional insight. This can be seen, for example, in the model where event "complaint must be archived" is followed by the function "archiving system". This issue stems from a strict alternation of events and functions which is often mentioned as a syntax requirement for EPCs, even though semantics formalizations (see e.g. [47]) do not require this alternation. Motivated by **G1** and reasoning that most of the events do not add much communicative value here, we remove the superfluous ones.

In the original model, there is an XOR-connector with a high degree of six (topmost connector in Fig. 2). At the connector's input side, it merges three alternative ways in which complaints enter the department. At its output side, it splits the further processing into three alternative routes. In the spirit of **G2**, the same routing logic is expressed in the transformed model with two subsequent connectors, both of a lower degree.

The original model has three different starting points and two different ending points. This is problematic as it is not directly obvious what the start and end conditions are. By adhering to **G3**, only a single start and a single end remain in the transformed model. Note that after the earlier application of **G1**, these are the only remaining events.

The part of the initial EPC that specifies the referral of a complaint is modeled in an unstructured way such that the routing is difficult to understand. There are alternative routes for internal and external referrals, each of which spawns off two concurrent routes. But to exploit the fact that *either* type of referral must be archived anyway, a sequence of logical connectors is used that is not nested. Following **G4** we represent the same logic in a structured way. Now, the archiving for internal and external referrals is modeled *within* each of the alternative paths. Even though this modification leads to a somewhat larger model, we gain in terms of structuredness since a larger part of the model now has properly nested connectors.

At the right branch, the original model uses an OR-join. When a complaint is handled immediately and not referred, the procedure requires that (i) the complainant must be contacted, (ii) the complaint must be archived, and (iii) there is an optional follow-up that needs to take place. Two of the three paths leading to the OR-join in Fig. 2 need to be synchronized. On the basis of **G5** this OR-join is removed and replaced by an equivalent but more readable construct.

The labels of the events and functions in the original model tend to differ in grammatical style and, in general, are quite long. For example, two functions at the top are labelled "complaint to be written down with form AZ2" and "Register receipt date of complaint letter". Inspired by **G6**, the alternatives "Write down complaint" and "Register letter receipt date" are used in the transformed model. Whether essential information is lost in this way, e.g. by not mentioning the specific form that must be used, depends on the exact purpose of the model and should be decided contextually.

The **G7** decomposition recommendation is not applicable to the original model, as its number of modeling elements, i.e. functions, events, and connectors together, is already below 50. In fact, the overall number decreases from 37 in the original model to 31 in the transformed one, in particular as a result of the application of **G1**.

It is important to note that the application of 7PMG does not touch the logic that is behind the original model. In fact, both models have the same behavior modulo branching bisimulation [45]. Both EPCs can be automatically translated to a transition system capturing the precise behavior and these transition systems are bisimilar, i.e., any state or sequence of actions in one model can be mimicked by the other model, and vice versa. Note that in this case we abstract from silent steps (e.g., invisible actions related to the handling of superfluous events) and unify the naming of functions in both models. Although the behavior did not change by restructuring, renaming, and reducing the original model, (i) it has become more understandable to humans and (ii) the risk is reduced that errors are introduced when it is modified or extended.

4. Prioritizing guideline

In the previous section, the application of 7PMG has been illustrated by the individual application of each of its elements. An important remaining issue is how to deal with situations where various guidelines are applicable *at the same time* but guide the modeler towards *different* directions. For instance, while fewer elements make a model more understandable (**G1**), reducing the degree of routing paths per element (**G2**) may actually require an increase of model elements. This can be seen in the example of the previous section, where the application of **G2** results in the addition of another connector (see Fig. 3). Clearly, there is a need for sensible priorities in applying the guidelines of 7PMG.

It should be noticed that the potential interaction effects between the seven proposed guidelines are intricate and diverse. For a give process model, many guidelines can be applicable, at various places in a process model, and conflicting to different degrees. A comprehensive prioritization that is both theoretically motivated and empirically validated is out of scope for this paper. Nonetheless, to arrive at some guidance for prioritizing the application of 7PMG's elements, we have taken the following approach. We contacted professional process modelers in our network, both in Germany and in the Netherlands, to invite them to participate in a workshop with us. The purpose was to (a) discuss the guidelines and (b) to establish a priority scheme on the basis of their expert opinions. In the first workshop, seven modelers from the German community of practice Berliner BPM-offensive participated; in the second workshop, 14 modelers joined from the Dutch practice of a major consultancy firm. These 21 professionals had an average experience of 5 years with process modeling and created over 50 process models during this period, indicating a considerable level of expertise.

Both workshops were carried out in exactly the same way. First, a presentation was given by the researchers on 7PMG,

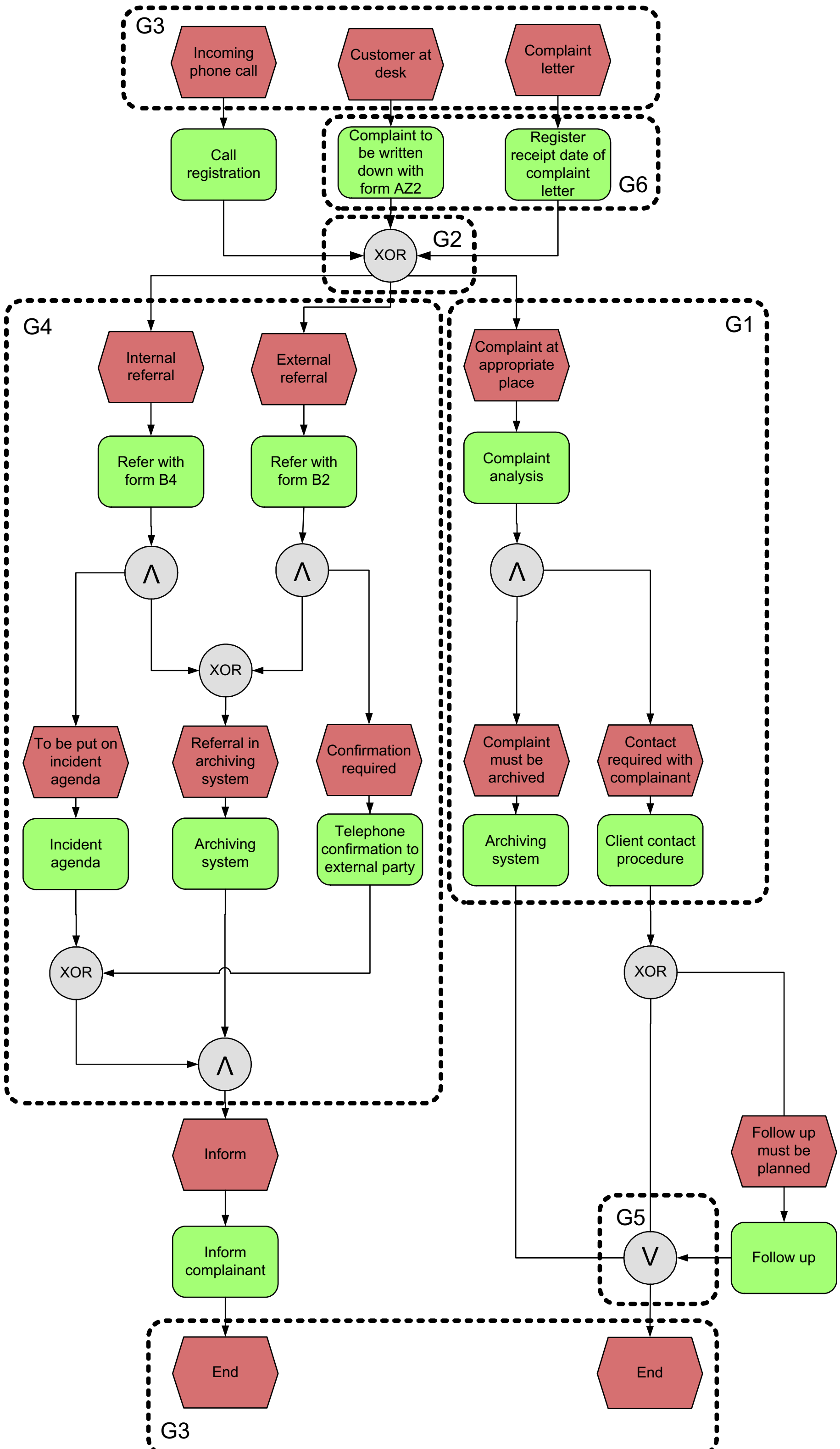

Fig. 2. The original complaint process model including markers.

explaining each guideline and addressing questions on its exact application. Next, all participants were asked to rank the seven guidelines with respect to their potential to improve the understanding of a process model using a scale of 1–7. For this scale, a rank of 1 indicates that a guideline is perceived as having the highest relative potential for improving understanding; a rank of seven indicates the other end of the scale, representing the lowest relative potential. The rationale behind the described inquiry is that a priority scheme can be inferred from such a ranking in case of conflicting guidelines. After all, a guideline which

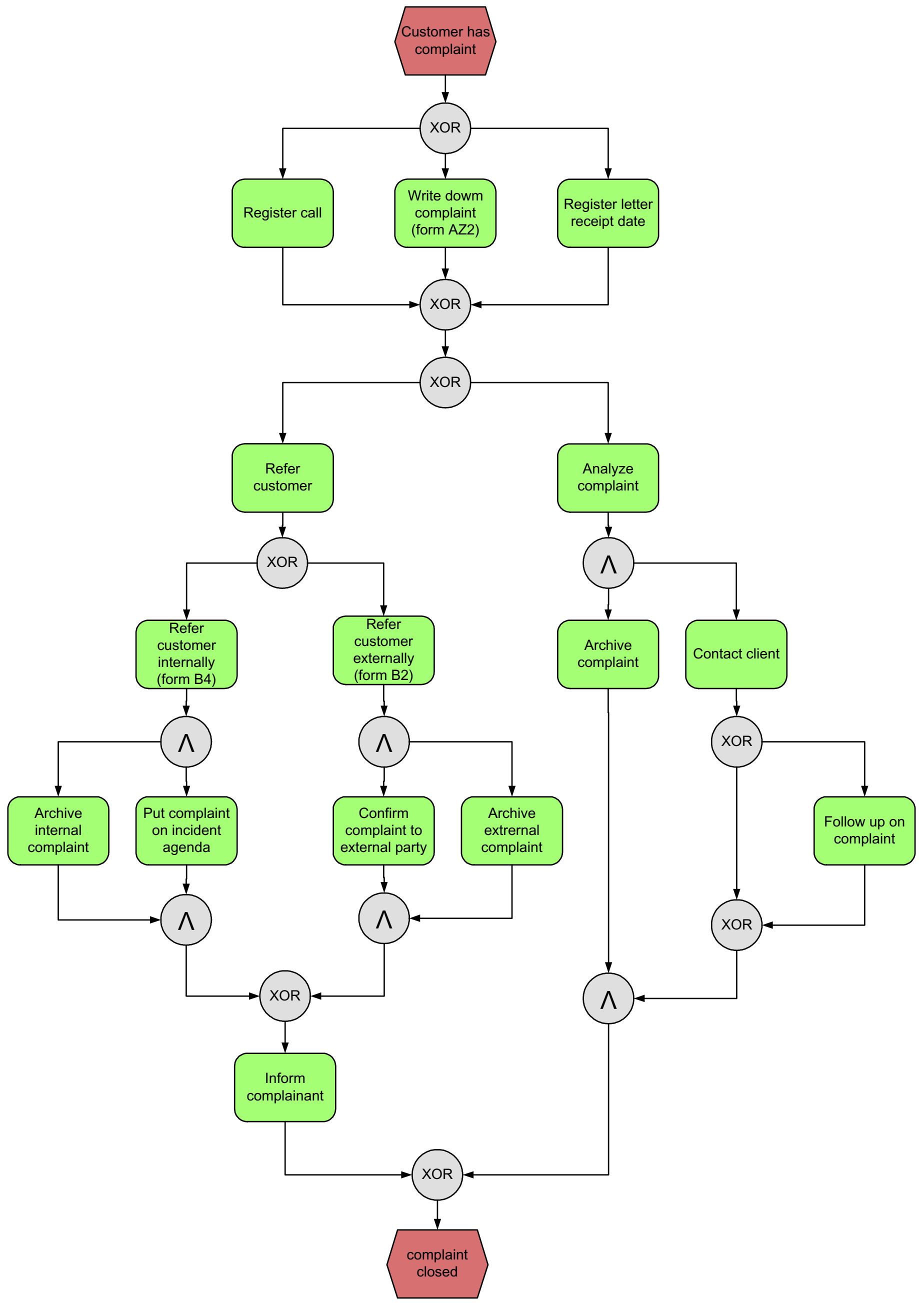

Fig. 3. The adapted process model.

Table 2
Prioritizing guideline 7PMG.

Position	No.	Explanation	Accumulated rank
1	**G4**	Model as structured as possible	58.5
2	**G7**	Decompose a model with more than 50 elements	73.5
3	**G1**	Use as few elements in the model as possible	80.5
4	**G6**	Use verb-object activity labels	84
5	**G2**	Minimize the routing paths per element	86.5
6	**G3**	Use one start and one end event	101
7	**G5**	Avoid OR routing elements	104

is generally perceived as having a big potential may be preferred over a guideline that is considered to have less potential. In ranking the guidelines, participants were not required to give a full ordering, i.e. assigning an equal rank to two or more guidelines was allowed. The important constraint that was taken into account was that the sum of the assigned ranks for each participant always equalled 28 (=1 + 2 + 3 ⋯ + 7).

To determine the prioritizing guideline all assigned ranks were accumulated per element of 7PMG, leading to results as shown in Table 2. In this table, it can be seen that **G4** is considered as the guideline with the most potential to improve a process model's understandability, as it has an accumulated rank total of 58.5. In contrast, **G5** is considered to have the least potential to do so, with an accumulated rank of 104. All other guidelines received ranks such that their accumulation was between these extremes.

The suggested use of this ranking is that a wide application of guidelines with higher positions should be favored over the wide application of conflicting guidelines at lower positions. For the example that was used earlier in this section – choosing between reducing the number of elements to make a model more understandable (**G1**) or adding connectors to lower the degree of routing paths per element (**G2**) – this means that one should restrict oneself to applying **G2** to extreme cases only, i.e. only those connectors with unusually high degrees. In other words, an extensive application of **G2** should be considered as an interference with the higher impact that can be expected from applying guideline **G1**.

It is worth mentioning that a side result from these workshops was a very lively discussion on process modeling guidelines with the professional modelers. While most of the elements of 7PMG were broadly recognized and some of them already consciously applied by the participants – in particular with respect to a structured modeling approach, cf. **G4** – additional heuristics were also brought forth that seem worthwhile to investigate in more detail. In particular, to either model from left to right or from top to bottom was mentioned most often as a heuristic not covered by 7PMG. This heuristic clearly relates to *layout* of a process model, the importance of which we already hypothesized about in earlier work on process model understandability [31].

5. Discussion

In this section we investigate 7PMG from different angles. First, we turn to some limitations before we reflect on its potential. Then, we consider 7PMG and its specific relationship to process modeling techniques and tools.

5.1. Limitations

Even though 7PMG is stronger in its foundations and more practical than many existing guidelines, it has some limitations that need to be reflected upon. The first that we want to highlight relates to 7PMG and *validity*: 7PMG does not relate to the content of a process model, but only to the way this content is organized and represented. Formal research has introduced various notions of behavior equivalence that can exist for process models of differing structures. 7PMG suggests ways of organizing such a structure of the process model while keeping its content intact. As the validity of a process model is and remains of the utmost importance, 7PMG complements this concern with recommendations for how to model. In this way, 7PMG does not help with the pragmatic question of what should be put into a model. This question still has to be assessed based on the purpose of modeling.

The second limitation of 7PMG relates to the prioritizing guideline that was described in Section 4. Clearly, the derived ranking has a small empirical basis as it relies on the involvement of 21 process modelers only. This could be seen on the one hand as a need for a wider involvement of process modelers' experience, but it also raises the question what alternative approaches may be available to arrive at a prioritizing guideline, e.g. through experimentation. Also, the prioritizing guideline that can be inferred from Table 2 is too coarse to be applicable at the micro level of, for example, deciding between the application of two conflicting guidelines. Still, it seems less attractive to focus at this stage on developing more fine-grained guidance for applying 7PMG. First of all, it does not seem realistic that guidance can be developed that is both conclusive and valid for all scenarios. Secondly, we expect the set of 7PMG to be extended as insights into process model quality develop over time. For example, the workshops that we carried out with the professional modelers have already provided us with inspiration for the investigation of further guidelines.

5.2. Potential

We continue this section with discussing the potential of 7PMG beyond the application that is hitherto sketched. If we suppose the availability of a function that quantifies the quality of a model (e.g. in terms of error probability) then this bears the potential of *automating the improvement* of a model. Several techniques for graph matching and graph edit distance calculation can potentially be applied when such an available regression function is used as a goal function. An approach like this would require further formal research on how behavior-preserving sets of edit operations can be defined that are complete. Some work has been done in this area by identifying change operations [50]. Follow up on this type of research might eventually provide mechanisms to automatically enforce 7PMG in a modeling tool and offer intelligent support for modifications.

The seven guidelines also point to some potential for quantifying the *competence* of process modelers by measuring the structure of models they create. This might base on the assumption that good modelers will intuitively put requirements into a model in such a way that the guidelines are followed to a large extent. On the other hand, competent modelers are likely to understand those models that deviate from the guidelines too. In an experiment that involved students from three European universities we have observed significant differences in performance with respect to reading process models. Although process modeling was part of the curriculum of each of the three groups, the amount of time spent on this subject differed. Students that were trained longer in process modeling and analysis, notably with respect to detecting deadlocks and recognizing block structures, were also better able to understand the process models that they were faced with [31]. In this way, 7PMG might offer ways to identify the difficulty of a model understanding task, and accordingly a way to assess how competent a person is in understanding the more difficult models.

5.3. Modeling techniques and modeling tools

There are additional matters of process modeling that are not directly addressed by the seven guidelines, but that are very closely related to it. We single out the subjects of (1) the *modeling technique*, i.e. the notation or language that is used to create a process model, and (2) the *modeling tool*, i.e. the software package that supports the use of a particular modeling notation hard return. Over the last decades, many different process modeling *techniques* have been proposed. Vendors and standardization bodies have a tendency to come up again and again with new diagramming techniques, yet hardly ever providing any empirical or theoretical validation. An approach that builds on empirical insights such as 7PMG would be desirable in this area, too. Some research has been conducted on comparing different process modeling languages from an end-user perspective. In particular, languages with an explicit representation of routing elements seem to be easier to understand [35]. These languages also tend to have less elements: In an EPC an AND-split requires only one connector while in a Petri net one needs to model a transition with various output places. In this ways, guideline **G1** and its supporting research [41,44] have some implications also on the language level: Those languages that require less elements representing the same fact than another language might be preferable for communication purposes. In fact, this is backed up by a more general insight that is claimed to relate to *all* forms of representations, i.e. that the ones that provide smaller models have the higher efficacy [51]. At the same time, a notation will *never* free the designer from the difficult work of actually mapping the real-world process onto the model.

The selection of a particular modeling technique determines to some extent what modeling *tool* to use, and vice versa. Some frameworks, like the cognitive dimension framework, stress that both should be studied jointly to understand how the user interacts with the notation via a tool [52]. An additional selection criterion, based on the guidelines we discussed, would be to favor modeling tools that further support the modeling process. For example, tools could warn users if their model becomes too large (**G1**) or if it contains connectors with a large degree (**G2**). Some existing technologies, like Windows Workflow Foundation[1] or the Yasper editor[2] already assist users in keeping their process model structured (**G4**). We have implemented an approach to support workflow design based on coupling and cohesion metrics in the past [53] and plan to rework it towards 7PMG support. Further functionality is discussed as part of the work on change patterns [50]. It is also important that a modeling tool allows for different views, e.g. the user should be able to dynamically choose the preferred level of granularity and select a combination of perspectives without actually changing the model.

6. Conclusion

In this paper we have addressed the mismatch between abstract recommendations for process modeling and technical insights into modeling practice. We consolidated prior empirical research and derived seven process modeling guidelines, resulting in 7PMG. In contrast to guidelines that exist in practice, each of our guidelines builds on a strong research foundation. In contrast to other research on process model quality, the guidelines are simple enough to be easily understood my modelers. In this way, our guidelines address the practical problem that many modelers in large industry projects require intuitive guidance. This fact is in particular emphasized by the low level of competence of casual modelers [6] and the high error rates (between 10% and 20%) in industry model collections [15].

Beyond these merits, we have also discussed some limitations of 7PMG. Most importantly, the guidelines give directions for which modeling alternative should be chosen. Yet, they do not directly help to make the trade-off between potentially contradicting rules. While the guidelines abstract from these problems, there are some solutions already available in current research to tackle this issue. For instance, the regression function derived in [44] can be maximized by using graph edit operations that preserve the behavior of the model. Such solutions have not yet been discussed in detail. They do belong to our agenda for future research.

Another important aspect of future research relates to the usability of the seven guidelines. In [24] the authors present findings from validating the SEQUAL quality model regarding its applicability. A similar approach can be considered here. In particular, standard survey designs from the information systems research field building on usefulness and ease of use perceptions can be adapted here to evaluate the practical merit of 7PMG.

A final observation that seems worth to make is that the large interest in industrial practice in process modeling is picked up and mirrored in recent years by widespread activities of academics in this field. In fact, important insights have accumulated in academia on when designers make errors and what kind of constructs are difficult to understand. Unfortunately, these insights are not used in current practice, which at this time leaves both communities disconnected. In that regard, we hope that 7PMG can serve as a tool for knowledge transfer, translating research findings into a concise yet concrete set of guidelines for the day-to-day practice of process modelers. In turn, academics may want to take their inspiration from the real-life problems that process modelers face.

[1] See http://en.wikipedia.org/wiki/Windows_Workflow_Foundation, last checked on December 11 2008.

[2] See http://www.yasper.org, last checked on December 11 2008.

References

[1] Y. Wand, R. Weber, Research commentary: information systems and conceptual modeling – a research agenda, Information Systems Research 13 (4) (2002) 363–376.
[2] I. Davies, P. Green, M. Rosemann, M. Indulska, S. Gallo, How do practitioners use conceptual modeling in practice?, Data and Knowledge Engineering 58 (3) (2006) 358–380
[3] R. Hung, Business process management as competitive advantage: a review and empirical study, Total Quality Management and Business Excellence 17 (1) (2006) 21–40.
[4] C. Russo, 12 Rules to make your ISO 9000 documentation simple and easy to use, Quality Progress 30 (3) (1997) 51–54.
[5] A. Dandekar, D. Perry, L. Votta, Studies in process simplification, Software Process: Improvement and Practice 3 (2) (1997) 87–104.
[6] M. Rosemann, Potential pitfalls of process modeling: part a, Business Process Management Journal 12 (2) (2006) 249–254.
[7] J. Gulla, T. Brasethvik, On the challenges of business modeling in large-scale reengineering projects, in: P.P. Chen, D.W. Embley, J. Kouloumdjian, S.W. Liddle, J.F. Roddick (Eds.), Fourth International Conference on Requirements Engineering, IEEE, Schaumurg, 2000, pp. 17–26.
[8] H.A. Reijers, R.S. Mans, R.A. van der Toorn, Improved model management with aggregated business process models, Data and Knowledge Engineering 68 (2) (2009) 221–243.
[9] J. Krogstie, G. Sindre, H. Jørgensen, Process models representing knowledge for action: a revised quality framework, European Journal of Information Systems 15 (1) (2006) 91–102.
[10] J. Becker, M. Rosemann, C. Uthmann, Guidelines of business process modeling, in: W. van der Aalst, J. Desel, A. Oberweis (Eds.), Business Process Management. Models, Techniques, and Empirical Studies, Springer, Berlin, 2000, pp. 30–49.
[11] B. Silver, Ten tips for effective process modeling, Bpms watch, BPMInstitute.org, <http://www.bpminstitute.org/articles/article/article/bpms-watch-ten-tips-for-effective-process-modeling.html>, Wednesday January 30, 2008.
[12] M. Koubarakis, D. Plexousakis, A formal framework for business process modelling and design, Information Systems 27 (5) (2002) 299–319.
[13] S. Sadiq, M. Orlowska, W. Sadiq, Specification and validation of process constraints for flexible workflows, Information Systems 30 (5) (2005) 349–378.
[14] D. Moody, Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, Data and Knowledge Engineering 55 (3) (2005) 243–276.
[15] J. Mendling, Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness, Lecture Notes in Business Information Processing, vol. 6, Springer, 2008.
[16] M.D. Zisman, Use of production systems for modeling asynchronous concurrent processes, Pattern-directed Inference Systems (1978) 53–68.
[17] C. Ellis, G. Nutt, Office information systems and computer science, ACM Computing Surveys 12 (1) (1980) 27–60.
[18] G. Keller, M. Nüttgens, A.-W. Scheer, Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)", Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1992.
[19] B. Curtis, M. Kellner, J. Over, Process modeling, Communications of the ACM 35 (9) (1992) 75–90.
[20] D. Georgakopoulos, M. Hornick, A. Sheth, An overview of workflow management: from process modeling to workflow automation infrastructure, Distributed and Parallel Databases 3 (1995) 119–153.
[21] S. Jablonski, C. Bussler, Workflow Management: Modeling Concepts, Architecture, and Implementation, International Thomson Computer Press, London, UK, 1996.
[22] A.-W. Scheer, ARIS Business Process Modelling, Springer-Verlag, 2000.
[23] O. Lindland, G. Sindre, A. Sølvberg, Understanding quality in conceptual modeling, IEEE Software 11 (2) (1994) 42–49.
[24] D. Moody, G. Sindre, T. Brasethvik, A. Sølvberg, Evaluating the quality of process models: empirical testing of a quality framework, in: S. Spaccapietra, S.T. March, Y. Kambayashi (Eds.), Conceptual Modeling – ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7–11, 2002, Proceedings, Lecture Notes in Computer Science, vol. 2503, Springer, 2002, pp. 380–396.
[25] I. ISO, Information technology – software product evaluation – quality characteristics and guide lines for their use, Iso/iec is 9126, 1991.
[26] A.S. Gücegliolu, O. Demirörs, Using software quality characteristics to measure business process quality, in: W. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management, Third International

Conference, BPM 2005, Nancy, France, September 5–8, 2005, Proceedings, Lecture Notes in Computer Science (LNCS), vol. 3649, Springer-Verlag, pp. 374–379.
[27] G. Canfora, F. García, M. Piattini, F. Ruiz, C. Visaggio, A family of experiments to validate metrics for software process models, Journal of Systems and Software 77 (2) (2005) 113–129.
[28] J. Cardoso, Process control-flow complexity metric: an empirical validation, in: Proceedings of IEEE International Conference on Services Computing (IEEE SCC 06), Chicago, USA, September 18–22, IEE Computer Society, 2006, pp. 167–173.
[29] J. Mendling, M. Moser, G. Neumann, H. Verbeek, B. Dongen, W. van der Aalst, A Quantitative Analysis of Faulty EPCs in the SAP Reference Model, BPM Center Report BPM-06-08, BPMCenter.org, 2006.
[30] J. Mendling, Detection and prediction of errors in epc business process models, Ph.D. thesis, Vienna University of Economics and Business Administration, http://wi.wu-wien.ac.at/home/mendling/publications/Mendling%20Doctoral%20thesis.pdf, 2007.
[31] J. Mendling, H.A. Reijers, J. Cardoso, What makes process models understandable?, in: G. Alonso, P. Dadam, M. Rosemann (Eds.), Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24–28, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4714, Springer, Brisbane, Australia, 2007, pp. 48–63.
[32] J. Mendling, M. Strembeck, Influence factors of understanding business process models, in: W. Abramowicz, D. Fensel (Eds.), Proceedings of the 11th International Conference on Business Information Systems (BIS 2008), Lecture Notes in Business Information Processing, vol. 7, Springer-Verlag, 2008, p. 142153.
[33] M. Rosemann, J. Recker, M. Indulska, P. Green, A study of the evolution of the representational capabilities of process modeling grammars, in: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006, Luxembourg, June 5–9, 2006, Proceedings, Lecture Notes in Computer Science, vol. 4001, Springer, 2006, pp. 447–461.
[34] R. Agarwal, A. Sinha, Object-oriented modeling with UML: a study of developers' perceptions, Communications of the ACM 46 (9) (2003) 248–256.
[35] K. Sarshar, P. Loos, Comparing the control-flow of epc and petri net from the end-user perspective, in: W. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management, Third International Conference, BPM 2005, Nancy, France, September 5–8, 2005, Proceedings, LNCS, vol. 3649, 2005, pp. 434–439.
[36] J. Recker, A. Dreiling, Does it matter which process modelling language we teach or use? An experimental study on understanding process modelling languages without formal education, in: M. Toleman, A. Cater-Steel, D. Roberts (Eds.), 18th Australasian Conference on Information Systems, The University of Southern Queensland, Toowoomba, Australia, 2007, pp. 356–366.
[37] W. Bandara, Factors and measures of business process modelling: model building through a multiple case study, European Journal of Information Systems 14 (2005) 347–360.
[38] A. Sharp, P. McDermott, Workflow Modeling: Tools for Process Improvement and Application Development, Artech House Publishers, 2001.
[39] T. Malone, K. Crowston, G. Herman (Eds.), Organizing Business Knowledge: The MIT Process Handbook, The MIT Press, 2003.
[40] J. Mendling, H.A. Reijers, J. Recker, Activity Labeling in Process Modeling: Empirical Insights and Recommendations, Information Systems. URL: <http://eprints.qut.edu.au/19625/>.
[41] J. Mendling, H. Verbeek, B. Dongen, W. van der Aalst, G. Neumann, Detection and prediction of errors in EPCs of the SAP reference model, Data and Knowledge Engineering 64 (1) (2008) 312–329.
[42] J. Vanhatalo, H. Völzer, F. Leymann, Faster and more focused control-flow analysis for business process models through sese decomposition, in: B. Krämer, K.-J. Lin, P. Narasimhan (Eds.), Service-oriented Computing – ICSOC 2007, Fifth International Conference, Vienna, Austria, September 17–20, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4749, Springer, 2007, pp. 43–55.
[43] V. Gruhn, R. Laue, What business process modelers can learn from programmers, Science of Computer Programming 65 (1) (2007) 4–13.
[44] J. Mendling, G. Neumann, W. van der Aalst, Understanding the occurrence of errors in process models based on metrics, in: R. Meersman, Z. Tari (Eds.), OTM Conference 2007, Proceedings, Part I, Lecture Notes in Computer Science, vol. 4803, Springer, 2007, pp. 113–130.
[45] R. Glabbeek, W. Weijland, Branching time and abstraction in bisimulation semantics, Journal of the ACM 43 (3) (1996) 555–600.
[46] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros, Workflow patterns, Distributed and Parallel Databases 14 (1) (2003) 5–51.
[47] E. Kindler, On the semantics of EPCs: resolving the vicious circle, Data and Knowledge Engineering 56 (1) (2006) 23–40.
[48] J. Recker, J. Mendling, On the translation between BPMN and BPEL: conceptual mismatch between process modeling languages, in: T. Latour, M. Petit (Eds.), Proceedings of the CAiSE Workshops at the 18th Conference on Advanced Information Systems Engineering (CAiSE 2006), Luxembourg, 2006, pp. 521–532.
[49] J. Vanhatalo, H. Völzer, J. Koehler, The refined process structure tree, Data and Knowledge Engineering 68 (9) (2009) 793–818.
[50] B. Weber, M. Reichert, S. Rinderle-Ma, Change patterns and change support features – enhancing flexibility in process-aware information systems, Data and Knowledge Engineering 66 (3) (2008) 438–466.
[51] K. Whitley, Visual programming languages and the empirical evidence for and against, Journal of Visual Languages and Computing 8 (1) (1997) 109–142.
[52] T. Green, M. Petre, Usability analysis of visual programming environments: a 'cognitive dimensions' framework, Journal of Visual Languages and Computing 7 (2) (1996) 131–174.
[53] I. Vanderfeesten, H.A. Reijers, W. van der Aalst, Evaluating workflow process designs using cohesion and coupling metrics, Computers in Industry 59 (5) (2008) 420–437.