On the Cognitive Effectiveness of Routing Symbols in Process Modeling Languages

Kathrin Figl[1], Jan Mendling[2], Mark Strembeck[1], and Jan Recker[3]

[1] Vienna University of Economics and Business, 1090 Vienna, Austria
`{kathrin.figl,mark.strembeck}@wu.ac.at`
[2] Humboldt-Universität zu Berlin, 10099 Berlin, Germany
`jan.mendling@wiwi.hu-berlin.de`
[3] Queensland University of Technology, Brisbane QLD 4000, Australia
`j.recker@qut.edu.au`

Abstract. Process models provide visual support for analyzing and improving complex organizational processes. In this paper, we discuss differences of process modeling languages using cognitive effectiveness considerations, to make statements about the ease of use and quality of user experience. Aspects of cognitive effectiveness are of importance for learning a modeling language, creating models, and understanding models. We identify the criteria representational clarity, perceptual discriminability, perceptual immediacy, visual expressiveness, and graphic parsimony to compare and assess the cognitive effectiveness of different modeling languages. We apply these criteria in an analysis of the routing elements of UML Activity Diagrams, YAWL, BPMN, and EPCs, to uncover their relative strengths and weaknesses from a quality of user experience perspective. We draw conclusions that are relevant to the usability of these languages in business process modeling projects.

Keywords: Process modeling, cognitive analysis, UML, YAWL, BPMN, EPCs.

1 Introduction

Business process models play an important role for the documentation of organizational processes and for the specification of information systems requirements. These models are specified using graphical modeling languages, such as EPCs [7], UML Activity Diagrams [22], YAWL [1], or BPMN [21]. These languages provide sets of graphical constructs, together with rules for how to combine these constructs, to express graphically relevant aspects of business processes, such as the tasks that have to be performed, the actors that are involved in the execution of these tasks, relevant data, and, notably, the control flow logic that describes the logical and temporal order in which tasks are to be performed. One important aspect in the control flow logic of a business process is that processes often contain points where parallel or alternative paths might be taken, or where such paths merge. Such points characterize the convergence or divergence of process flows [9]. In process models, points of convergence or divergence

W. Abramowicz and R. Tolksdorf (Eds.): BIS 2010, LNBIP 47, pp. 230–241, 2010.

are typically expressed through *routing elements* such as "Gateways", "Connectors, or "Splits" and "Joins" [29]. While these routing elements are sometimes well-defined formally [9], they remain a key reason for modeling errors such as violation of deadlock and synchronization rules [16], and may further lead to understandability problems with practitioners [15].

In this paper, we discuss the representation of routing elements in different process modeling languages from a usability perspective. An important reference discipline to discuss usability issues of process modeling languages is cognitive psychology. This area deals with insights on how humans process information, create knowledge, and solve problems. A central property of the human brain is that it works with specialized areas. *Visual perception* enables understanding of sensory information, which is transmitted to the *sensory memory*, which holds an after-image of that information for a moment. This after-image is analyzed and key concepts are held in *short-term memory* for some minutes. Those aspects that can be linked to other prior knowledge can become part of *long-term memory*. All these three different memory parts have different characteristics, which are the foundation of various theories of comprehension, learning, and problem solving. In this paper, we argue correspondingly that these characteristics inform useful perspectives that can also be considered for discussing relative advantages and drawbacks of process modeling languages.

There are many orthogonal characteristics of process modeling languages that may be subject to a cognitive investigation. In this paper we draw on a recent framework proposed by Moody and Van Hillegersberg [18] that describes desirable properties of notation elements from a perceptual perspective. The aim of our paper is to discuss how the principles in [18] can be used to identify strengths and weaknesses of process modeling languages. Furthermore, we synthesize them in terms of propositions, which speculate on the cognitive effects of these strength and weaknesses. In this way, we pave the way towards an empirical investigation of their arguments in an experimental setting. Our research further identifies normative principles that can assist practitioners in the selection decision for a particular process modeling language.

We proceed as follows. In Section 2, we summarize the major arguments on usability of modeling languages and discuss cognitive theories which are relevant for understanding process modeling languages. Section 3 then turns to the work by Moody and Van Hillegersberg, which we utilize in Section 4 to discuss strengths and weaknesses of process modeling languages and identifies a number of propositions from this analysis. Finally, Section 5 concludes the paper and gives an outlook on future research.

2 The Cognitive Side to Process Modeling

Usability in general can be described as the measure of the ease of use and quality of the user experience when interacting with a device the user can operate in some way or another [19]. There are at least two areas in which these considerations also pertain to process modeling (see [6]):

- **Creating models:** Usability aspects of creating models with a given process modeling language include the effort (e.g. time) required to construct models, the subjective ease-of-use of doing so, and the ease-of-learning a language. The result of such a modeling process is a model, which can be assessed concerning different quality aspects (such as correctness, accuracy, detail, completeness, quality, type of errors).
- **Interpreting models:** In the process of model understanding, much cognitive activity is involved. The outcome of understanding is cognitive per se, i.e. it is created in the viewer's cognition and may only be measured by observing a viewer's problem-solving capacity, the level of process understanding generated from reading a process model or different recall capacities. Models from different modeling languages are likely to differ according to the effort required to interpret them and develop understanding, as well as in the perceived difficulty of obtaining information about a process through the visual representation chosen in a model.

Both model creation and model interpretation tasks highlight the complex interplay between human cognitive models and the visual models used to convey process information. The ability of a language to support appropriate translations between cognitive and visual models, therefore, is an essential criterion for determining the usability of any given modeling language.

We believe that Cognitive theories are central to our understanding of the usability of different modeling languages. Mental processes such as visual perception, information processing, reasoning and problem solving, attention, as well as short and long term memory are affected in learning how to use specific modeling languages, creating models, and understanding models. Our research, therefore rests on the assumption that there are specific aspects attributable to process modeling language that are likely to have an impact on the cognitive processes of the individual working the language.

Process models represent complex organizational relationships in a visual diagram; yet, humans have limited information processing capabilities (see [30]). Therefore, a main goal in the design of process modeling languages is to reduce *cognitive load* for users to enable more effective problem solving. Low cognitive effort is positively related to a variety of quality aspects of models, such as perceived ease of understanding [12]. Cognitive load is determined by the amount of elements needed to be paid attention to at a point of time. There is a natural limit of the capacity of short-term memory of humans of approximately 7 +/- 2 elements [17], which, in consequence, should be a criterion for selecting an appropriately parsimonious symbol set in any process modeling language.

Dual coding theory further suggests that humans' short-term (or working) memory [3] includes a phonological loop and a visuo-spatial sketchpad. The theory postulates that visual information (e.g. graphical elements in process models) and verbal information (e.g. textual labels) are stored and processed differently via separate mental channels that do not compete with each other [23]. The cognitive multimedia learning theory [13] proposes that, as per the *contiguity principle*, understanding of graphical material (such as a process model) is better, when text and pictures are presented spatially near each other. In consequence, text and icons belonging together should be placed near each other in process models. We note that such a situation is only partially, and inconsistently, given in process modeling, where graphical elements (e.g., routing elements) may be, but don't have to be, annotated with textual labels to convey additional semantics about the elements.

The cognitive load theory [28] further details how the working memory load influences learning and knowledge acquisition. The theory differs between three types of cognitive load: *intrinsic*, *extraneous* and *germane cognitive load*. In contrast to *intrinsic cognitive load* (which is determined by the complexity of information, i.e. the amount of elements, and their relations and interactions), the *extraneous cognitive* load is influenced by the way the information is represented [10]. While the cognitive load devoted to learning and understanding (*germane cognitive load*) should be promoted, *extraneous cognitive load* should be held low. This can, for example, be achieved by reducing additional, irrelevant information.

By mapping the cognitive load theory to the context of process modeling, it becomes clear why modeling languages might vary in their cognitive effectiveness. If the same information is modeled in different modeling languages, the resulting models should, to a large extend, have a similar *intrinsic cognitive load*, but they differ in their *extraneous cognitive load*. The amount of *extraneous cognitive load* caused by the modeling languages leads to differences in learning and understanding (see, e.g., [5]).

Due to the complex control flow logic attributed to organizational processes, the creation and understanding of process models is likely to demand high cognitive reasoning and effort for logical thinking for human users. Visual models not only demand, but also support users in their reasoning processes because they convey cues to the next logical step in reasoning about a process-related problem by representing process information (e.g., tasks to be performed) in the context of adjacent locations (e.g., in the context of the routing elements that describe important business rules pertinent to the execution of the task).

Research has shown that there are systematic fallacies (so called 'illusory inferences') when individuals internally construct, or interpret mental models on premises including modeling-level connectives (like conjunctions, inclusive, or exclusive disjunctions) [8]. This situation may also be present for externalized visual process models. The body of literature on error analysis of process models suggests the existence of systematic reasoning fallacies concerning routing elements [15]. Since correct interpretation of routing elements in process models is inevitable for overall understanding of the process depicted, we thus conjecture that different visualisations of routing elements in different process modeling languages determine, at least partly, to which extent different process model languages support understandability and the quality of the user experience.

3 Evaluating the Cognitive Effectiveness of Modeling Languages

The form of visual information representation can have a significant impact on the efficiency of information search, explicitness of information, and problem solving (see [11]). One of the key goals for the visual design of a model is that viewers draw attention to those components crucial for understanding and cognitive inferencing [27]. According to Moody and Hillersberg [18] there are 5 principles that inform our understanding of the cognitive effectiveness of visual modeling languages: representational clarity, perceptual discriminability, perceptual immediacy, visual expressiveness and graphic parsimony:

- **Representational Clarity:** This principle points out the importance of a good fit between the graphical symbols used in a modeling notation and the semantic concepts they refer to. Anomalies like symbol redundancy (more than one symbol represents the same concept), overload (one symbol represents more than one concept), symbol excess and deficit (there are graphical symbols without a correspondence to a semantic construct or vice versa) should be avoided, since they lead to ambiguity and additional unnecessary cognitive load for the user [18]. A recent comparative analysis [25] compared the modeling languages EPC and BPMN concerning their representational completeness and clarity. The results of this comparison revealed numerous differences between process modeling languages that the authors expect to have an effect on the cognitive efficiency of the languages considered.
- **Perceptual Discriminability:** Perceptual discriminability of symbols determines how easy it is for a user to distinguish between and visually recognize different symbols in a graphical artefact such a model. It is highly influenced by the amount of visual variables in which symbols differ (referred to as visual distance). If visual symbols are highly unique on a visual variable, they are likely to 'pop out' and are easy to locate in a model [18]. Low perceptual discriminability can lead to misunderstandings. Research showed that for instance rectangles and diamonds in ER diagrams are easily confused (see [20]). On the other hand, if different symbols in a notation have similar attributes as color or shape, they are likely to be recognized as belonging together. In consequence, symbols in a modeling language should differ sufficiently by visual variables to be perceptual discriminable. However, sometimes it is intended that symbols share visual variables if they should be recognized as related to each other.
- **Perceptual Immediacy:** Perceptual immediacy supports the user's understanding of the meaning of graphical symbols and representations, and describes whether symbols and their corresponding concepts are easily associated. Icons, for example, are easily associated with their referent real-world concepts. Iconic representations for classes of activities could improve the understandability of process models [14], but are not yet commonly used. Additionally, spatial relationships of symbols can help to induce specific desired interpretations by a user (e.g. left-to-right implies sequence). Even a

small layout change in the same graph may transport a different meaning (e.g. centrality vs. hierarchy) (see [2]). Modeling languages are likely to differ according to the intuitiveness of the visual metaphors they use for expressing real-world concepts. Specifically, modeling effectiveness and efficiency will be higher if the symbols used in a modeling notation are more similar to the concept of node-relationship arc depiction of information [4]. Therefore, it can be hypothesised that process modeling languages with higher levels of nodes and edges are likely to be intuitively understandable because of their compatibility with internal mental representations.

- **Visual Expressiveness:** Modeling notations which fully exploit the range of visual variables (spatial dimensions like horizontal and vertical, as well as shape, size, colour, brightness, orientation, and texture) have higher visual expressiveness. In comparison to a textual representation (words), which are encoded verbally in their reading direction, visual symbols are internally encoded in their spatial arrangement (see [26]). Therefore, it is of importance to assess spatial dimensions of modeling notations. Using swimlanes in activity diagrams, for example, includes both planar variables for depicting information on who is working on an activity (see [18]).
- **Graphic Parsimony:** Parsimony is an indicator of graphical model complexity and is determined by the use of embedded symbols and distinct graphic symbols [20]. High graphic complexity of a modeling language, e.g., the use of too many constructs and routing elements in one model, may impair understanding and therefore should be avoided [15].

4 Language Differences

4.1 Basic Elements of Process Modeling Languages

In general, process modeling languages include a number of basic, generic, and consensual elements that allow to define process flows. Typically, these languages provide symbols for different variants of task nodes, events, or start and end of the process. A task node models a clearly defined action or process step. Tasks can be conducted by human users or software agents. Below we briefly introduce the routing elements, before the subsequent sections discuss how different process modeling languages define these generic elements.

Split: A split node models parallel branches in a process. Thus, sub-processes started by the same split node are performed simultaneously.

Join: A join node synchronizes parallel sub-processes. Typically, a join node synchronizes sub-processes resulting from a split node.

Decision: A decision node models choices in a process flow. Decision nodes thereby define alternative routing options in a certain process.

Merge: A merge node consolidates different (optional) paths that result from a choice made by a decision node.

Figure 1 shows four symbol sets of routing elements of EPCs, UML, YAWL, and BPMN, respectively. The figure shows splits and joins as well as decision and merge from left to right.

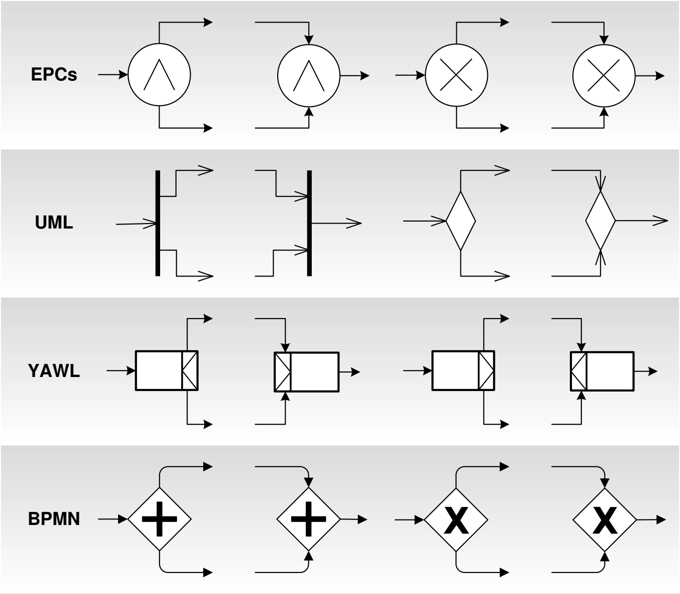

Fig. 1. The routing elements of EPCs, UML, YAWL, and BPMN

4.2 Event-Driven Process Chains (EPCs)

The EPC notation contains three types of symbols: circles for connectors, rounded rectangles of functions, and hexagons for events. Different connectors can be distinguished by their symbol, × for XOR, ∧ for AND, and ∨ for OR.

As per our cognitive effectiveness criteria introduced above, we observe that the use of EPCs may incur usability problems for the users working with the language. Specifically, we identify a problem to easily discriminate AND and OR connectors. They share the same symbol, albeit mirrored vertically. Altogether, the symbol set of EPCs is very abstract, such that there is little immediacy and intuition of the symbols. Visual expressiveness is also limited. Some tools depict events and functions in red and green color, which helps to distinguish them. The notation is also parsimonious by using only a small symbol set, yet, the parsimony of the language limits its representational clarity as many of the symbols are overloaded from an ontological perspective [25].

4.3 UML Activity Models

The main element of an activity model is an Activity. The Activity represents a process that consists of Actions and different types of control nodes. Actions thereby define the tasks (steps) that are performed when executing the corresponding Activity. To model the (synchronous or asynchronous) invocation of other processes from an Activity, one may use an action symbol (round-cornered rectangle) including a small rake symbol. In essence, this modeling element defines that the corresponding step in a process is itself a process that is modeled via an own Activity model.

Activity models have a token semantics, similar (but not equal) to Petri nets. A Decision node (a routing element) is represented by a diamond-shaped symbol that has one incoming and multiple outgoing edges. A Merge node is represented by a diamond-shaped symbol and has multiple incoming and one outgoing edge. A Fork node is represented via a thick line and has one incoming and multiple outgoing edges. A Join node is represented via a thick line and has multiple incoming and one outgoing edge.

The UML notation partially violates some of the criteria we consider in our evaluation. There is some gap of representational clarity with different end symbols (activity final and flow final). On the other hand, the different end symbols have clearly defined and distinct semantics. A strong point of UML is the clear discrimination between decisions and concurrency - two important aspects of routing elements. Indeed, both routing elements are significantly different. Visual expressiveness is used to distinguish concurrent from alternative branching. Altogether, the notation set is quite parsimonious whilst being reasonably expressive.

4.4 Yet Another Workflow Language (YAWL)

YAWL uses a unique start and end condition to mark the beginning and termination of a process. Conditions are used to separate tasks (similar to places in Petri nets), yet they do not need to be explicitly drawn. Tasks are depicted as rectangles. Routing behavior is defined by thin blocks on the left-hand and right-hand side of tasks. Both XOR and AND splits and joins use triangle symbols.

On the basis of our criteria, we can point to a number of areas with potential usability concerns pertaining to the YAWL language. Representational clarity is violated, because a decision and a simple merge can be described either by conditions with multiple incoming and outgoing arcs, or with XOR joins and splits, thereby affording the problem of ontological redundancy. We further identify a problem with the visual discrimination of AND and XOR splits and joins as these routing elements share the same visual symbol. The semantics have to be identified based on the position on the split/join block relative to and alongside with the number of incoming and outgoing arcs to the respective task. YAWL further offers only limited visual expressiveness due to little variation in shape and size. On the other hand, we note the language to be relatively parsimonious, at least when compared to more recent languages such as BPMN.

4.5 Business Process Modeling Notation (BPMN)

Aside from a variety of elements to depict tasks and events in a process model, BPMN offers a set of gateways for specifying routing constraints. Splits and joins both use a diamond shape. XOR gateways can be drawn without a symbol or with an X inside. AND gateways contain a plus sign.

The BPMN language exhibits different weaknesses from a cognitive expressiveness perspective. Representational clarity is partially violated [24]. XOR gateways can be described with different symbols (blank diamond and X diamond). This may be confusing to model readers. Also, XOR gateways as X diamonds are difficult to distinguish from AND gateways with a plus. Intermediacy is rather low with the blank set of symbols. We further note a somewhat limited visual expressiveness for the standard set of symbols: tasks and gateways are both quadrangles and gateways are circles. Texture and color is not used in the standard symbol set, but is often introduced by modeling tools like ARIS, e.g., to highlight different types of tasks. Finally, there is hardly any graphic parsimony. BPMN uses an extensive set of symbols and subtype symbols. This might make it difficult to learn the language.

4.6 Evaluation and Discussion

Based on the number and types of issues identified above we can hypothesize about understanding and learning performance depending on the process modeling language. In Table 1 we conceptualize our analysis above by highlighting relatively good ("+"), average ("+/-") and weak support ("-")of the languages considered, in terms of their cognitive expressiveness of routing elements in process models.

Table 1. Relative strengths and weakness of routing element visualizations in different process modeling languages

	EPCs	UML	YAWL	BPMN
Representational Clarity	-	+/-	-	-
Perceptual Discriminability	-	+	-	-
Perceptual Immediacy	-	-	-	-
Visual Expressiveness	+/-	+	-	+/-
Graphic Parsimony	+	-	-	-

Representational Clarity: A lack of clarity is likely to result in two problems when creating a model based on redundant, overloaded or excess symbols. With representational redundancy, modelers may require more time to decide which among several alternative elements to use for representing routing elements relevant to a real-world process. Models created by different modelers may further vary significantly in terms of the symbols they use for the same routing decisions taken in a process. These problems are likely to show up with BPMN and YAWL. On the other hand, symbol overload may result in corresponding ambiguity for somebody who is reading the model. This issue seems to be more relevant to EPCs and Petri nets, due to the parsimony of the symbol set offered. Overall, we note that UML activity diagrams appear to be attributed the highest degree of clarity across the languages discussed.

Perceptual Discriminability: Good discrimination should support fast visual perception and good recall in short term memory. From our analysis, it appears that YAWL is very weak in discriminating routing elements. This is also partially the case for EPCs (OR versus AND) and BPMN (X versus +). UML very clearly distinguishes concurrency and alternative branching with significantly different symbols.

Perceptual Immediacy: Many language elements in the languages considered are very abstract such that no intuition is provided. We note that some languages (e.g. EPCs and BPMN) offer perceptual immediacy for events and tasks, yet the graphical representation of routing elements is a weakness inherent in all languages we consider in this paper.

Visual Expressiveness: The visual expressiveness of process modeling languages is limited. BPMN and EPCs offer markers to distinguish different types of routing behaviors, and UML uses two distinct graphical shapes albeit with no explicit markers, to distinguish concurrency from alternative routing.

Graphic Parsimony: BPMN is by far the richest language in terms of its symbol set. Therefore, it should also be the most time consuming to learn. YAWL and UML also provide a rather large set of symbols. EPCs are lean in terms of symbols, such that they should be easy to learn notation-wise. On the other hand, they are the most abstract notations such that they should be the most difficult ones to be applied correctly.

Forthcoming from this synthesis of our evaluation, we can uncover a set of dimensions that can be used for a comparative empirical evaluation of the different process modeling languages. A particular aspect that is strongly supported by one language and weakly by another should directly materialize in a difference in understanding or learning performance. For example, one may hypothesize that the superiority of UML in terms of perceptual discriminability may result in better interpretation or comprehension performance of individuals confronted with UML models, when compared to, say, EPCs or BPMN. A comparative empirical evaluation on basis of the evaluation presented may further identify the relative importance of the different cognitive effectiveness criteria, to explaining or predicting interpretation effectiveness or efficiency of different process modeling languages. It could turn out, for instance, that graphic parsimony is a more relevant criterion to establish interpretational efficiency, than, say, perceptual immediacy. At this stage of our research, we note that a respective empirical experiment is missing; however, we note the potential of our research to inform a research agenda for a series of controlled studies to test the predictions made.

5 Conclusion

Despite increasing consciousness about the need to consider the (non-technical) user's point of view in the area of information systems engineering, little research has been undertaken in order to improve and understand the usability of modeling languages. In this paper, we contribute to this emerging stream of research by presenting a set of cognitive effectiveness criteria informed by cognitive theories, such that we can identify differences in four process modeling languages in terms of the suspected usability of these languages for modeling processes. The main contribution of this paper is an analysis of the symbol sets of four prominent process modeling languages for the modeling of routing elements to depict the control flow logic of business processes. Our evaluation results uncovers differences between the languages according to representational clarity, perceptual discriminability, and graphic parsimony. Our research informs a series of experimental studies that may test the predictions made. Further, we can extrapolate our analysis to the symbol sets in the process modeling languages that are used to graphically express tasks, events, actors, data or other process-relevant facts.

We expect that our research contributes to our understanding of process modeling languages, and assists the development and selection of process modeling languages likewise.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. Information Systems 30(4) (June 2005)
2. Aranda, J., Ernst, N., Horkoff, J., Easterbrook, S.M.: A framework for empirical evaluation of model comprehensibility. In: 29th International Conference on Software Engineering (ICSE 2007), Minneapolis, USA (2007)
3. Baddeley, A.D., Hitch, G.: Working memory. In: The psychology of learning and motivation: Advances in research and theory, vol. 8, pp. 47–89. Academic Press, New York (1974)
4. Bajaj, A., Rockwell, S.: COGEVAL: A Propositional Framework Based on Cognitive Theories To Evaluate Conceptual Models. In: Siau, K. (ed.) Advanced Topics in Database Research, pp. 255–282. Idea Group Publishing, USA (2005)
5. Chandler, P., Sweller, J.: Cognitive load while learning to use a computer program. Applied Cognitive Psychology 10(2), 151–170 (1996)
6. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. Requirements Engineering 9(4), 248–260 (2004)
7. Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK) (1992)
8. Khemlani, S., Johnson-Laird, P.N.: Disjunctive illusory inferences and how to eliminate them. Memory & Cognition 37(5), 615–623 (2009)
9. Kiepuszewski, B., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Fundamentals of control flow in workflows. Acta Informatica 39(3), 143–209 (2003)
10. Kirschner, P.A.: Cognitive load theory: implications of cognitive load theory on the design of learning. Learning and Instruction 12(1), 1–10 (2002)
11. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11(1), 65–100 (1987)
12. Maes, A., Poels, G.: Evaluating quality of conceptual modelling scripts based on user perceptions. Data & Knowledge Engineering 63(3), 701–724 (2007)
13. Mayer, R.E.: Multimedia Learning. Cambridge University Press, Cambridge (2001)
14. Mendling, J., Recker, J., Reijers, H.A.: On the usage of labels and icons in business process modeling. International Journal of Information System Modeling and Design 1(2) (2010)

15. Mendling, J., Reijers, H., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information and Software Technology (2) (2010)
16. Mendling, J., van Dongen, B.F., van der Aalst, W.M.P.: Getting rid of or-joins and multiple start events in business process models. Enterprise Information Systems 2(4), 403–419 (2008)
17. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
18. Moody, D., Hillegersberg, J.: Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In: Gašević, D., Lämmel, R., Van Wyk, E. (eds.) SLE 2008. LNCS, vol. 5452, pp. 16–34. Springer, Heidelberg (2009)
19. Nielsen, J.: Usability 101: Introduction to usability (2009)
20. Nordbotten, J.C., Crosby, M.E.: The effect of graphic style on data model interpretation. Information Systems Journal 9(2), 139–155 (1999)
21. OMG. Business Process Modeling Notation (BPMN), Version 1.2, formal 2009-01-03, The Object Management Group (January 2009), `http://www.omg.org/spec/BPMN/1.2/`
22. OMG Unified Modeling Language (OMG UML): Superstructure, Version 2.2, formal 2009-02-02, The Object Management Group (February 2009), `http://www.omg.org/technology/documents/formal/uml.htm`
23. Paivio, A.: Dual coding theory: Retrospect and current status. Canadian Journal of Psychology 45(3), 255–287 (1991)
24. Recker, J., Indulska, M., Rosemann, M., Green, P.: How good is bpmn really? insights from theory and practice. In: Ljungberg, J., Andersson, M. (eds.) 14th European Conference on Information Systems, Goeteborg, Sweden, pp. 1582–1593. Association for Information Systems (2006)
25. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business process modeling- a comparative analysis. Journal of the Association for Information Systems 10(4), 333–363 (2009)
26. Santa, J.L.: Spatial transformations of words and pictures. Journal of Experimental Psychology: Human Learning & Memory 3, 418–427 (1977)
27. Scaife, M., Rogers, Y.: External cognition: how do graphical representations work? Int. J. Hum.-Comput. Stud. 45(2), 185–213 (1996)
28. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science: A Multidisciplinary Journal 12(2), 257–285 (1988)
29. Verbeek, H.M.V., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Verifying workflows with cancellation regions and or-joins: An approach based on relaxed soundness and invariants. The Computer Journal 50(3), 294–314 (2007)
30. Vessey, I.: Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*. Decision Sciences 22(2), 219–240 (1991)