

Making sense of business process descriptions: An experimental comparison of graphical and textual notations

Ottensooser, Avner; Fekete, Alan; A. Reijers, Hajo; Mendling, Jan; Menictas, Con

Published in:
Journal of Systems and Software

DOI:
[10.1016/j.jss.2011.09.023](https://doi.org/10.1016/j.jss.2011.09.023)

Published: 01/10/2012

Document Version
Peer reviewed version

[Link to publication](#)

Citation for published version (APA):

Ottensooser, A., Fekete, A., A. Reijers, H., Mendling, J., & Menictas, C. (2012). Making sense of business process descriptions: An experimental comparison of graphical and textual notations. *Journal of Systems and Software*, 85(3), 596 - 606. <https://doi.org/10.1016/j.jss.2011.09.023>

Making Sense of Business Process Descriptions: An Experimental Comparison of Graphical and Textual Notations

Avner Ottensooser

School of Information Technologies, University of Sydney, Australia

Alan Fekete

School of Information Technologies, University of Sydney, Australia

Hajo A. Reijers

School of Industrial Engineering, Eindhoven University of Technology, The Netherlands

Jan Mendling

Institute for Information Business, Wirtschaftsuniversität Wien, Austria

Con Menictas

Newcastle Business School, Faculty of Business and Law, The University of Newcastle, Australia

Abstract

How effective is a notation in conveying the writer's intent correctly? This paper identifies understandability of design notations as an important aspect which calls for an experimental comparison. We compare the success of university students in interpreting business process descriptions, for an established graphical notation (BPMN) and for an alternative textual notation (based on written use-cases). Because a design must be read by diverse communities, including technically-trained professionals such as developers and business analysts, as well as end-users and stakeholders from a wider business setting, we used different types of participants in our experiment. Specifically, we included those who had formal training in process description, and others who had not. Our experiments showed significant increases by both groups in their understanding of the process from reading the textual model. This was not so for the graphical model, where only the trained readers showed significant increases. This finding points at the value of educating readers of graphical descriptions in that particular notation when they become exposed to such models in their daily work.

[☆]JSS revised draft 01-04-2011

Email addresses: avner@it.usyd.edu.au (Avner Ottensooser), alan.fekete@sydney.edu.au (Alan Fekete), h.a.reijers@tue.nl (Hajo A. Reijers), jan.mendling@wu.ac.at (Jan Mendling), con.menictas@newcastle.edu.au (Con Menictas)

1. Introduction

In software engineering, early design decisions are typically captured in terms of models. Notations play an important role in this context since they define the rules of how models can be constructed. Agreement upon a particular notation is required before models can be used as a means of communication among stakeholders. Then, models that comply with the selected notation can be effectively used to convey domain details to system designers, as input to model-driven software development tools [21, 29], and for documenting the domain for future reference [17]. This central importance of notations is emphasized for instance in [70]:

Notations have been a part of software design since the beginning. Any time design thought is externalised, such thought must be written down in some structure or form that supports interpretation at a later time by others, oneself, or a computerised program. It is no surprise, then, that notations continue to serve as a primary driver of research in the community.

Business processes are of significant importance to the early design phase of a software development project since they provide a procedural view on the business. A fundamental problem in this context is the selection of an appropriate notation for defining business process models. Several *flowchart-like* languages are currently used in practice, including languages like Event-driven Process Chains (EPCs) [39], YAWL [1], or diagrams offered by the Unified Modelling Language (UML), most notably Activity Diagrams [62]. Recently, standardization efforts have led to the definition of the Business Process Model and Notation (BPMN) [47]. In contrast to that, the alternative of utilizing textual descriptions for capturing business processes exists, such as *written use cases* [48, 3, 8]. Up until now, it is unclear which type of representation should be preferred for capturing business processes. In other contexts there has been some work arguing for the superiority of graphical notations, and some that seems to favour textual forms. Our goal is an empirical examination of the issue for business process notations.

The preference for a particular type of notation must be based on suitable criteria. Many characteristics have been proposed to study and compare notations, including expressive power [62], formal analysis capabilities [32], terseness [71], aesthetics [73], and usability [70, 26]. The understandability of process models is of particular relevance for process-oriented design. It has been found that model understanding has a significant effect on process redesign success [28]. Furthermore, the choice of a process modelling notation appears to be a significant success factor of a process modelling project [6]. For these reasons, we focus on the understandability of different process modelling notations here. More specifically, we aim to address the gap of empirical research on the relative strengths and

weaknesses of textual and graphical notations for the description of business processes.

When evaluating how readers understand a notation, it is important to identify the population of readers being considered. We here consider two different kinds of reader, because business process descriptions might be examined by both business analysts (who are trained in process models) and by business users (who are not trained in modelling), as they are collaborating in the development of software. Indeed, some notations for modelling business processes were explicitly designed to communicate among members of a heterogeneous team like this [47]. We use students from different educational programs, as proxies for these two communities of readers.

In this paper, we present findings from an experiment we designed to answer the research question of whether textual or graphical notations provide for a better understanding of business processes. We conducted a between-grammar study that measured understanding performance of a business process based on a graphical and a textual representation as alternative treatments. The experiment was conducted with 196 participants from three different universities in the Netherlands, Germany, and Australia.

In the experiment, we use BPMN [47] as an example of a graphical notation for describing business processes, and the Cockburn format for written use cases [3, 8] as the textual notation. Both notations are well-suited for studying graphical and textual process descriptions. BPMN has recently become the de facto standard for graphical process modelling. It is supported by a plethora of free and commercial process modelling tools and has influenced other standardisation bodies such as the Workflow Management Coalition revised their standard development efforts to incorporate BPMN [56]. Indeed, BPMN integrates most of the concepts used by prior process modelling languages [58] and supports an extensive set of workflow patterns [75]. Still, all the above listed process modelling notations use a consensual core set of elements [38]. Also empirical indications exist that differences in terms of understandability are not significant when selecting one or another of them [63, 57]. The written use case notation [3, 8] is arguably one of the most widely employed requirement gathering notations in the industry, and it is accepted by both IT professionals and business managers [34, page 298]. Cockburn's format is among 28 use case notations surveyed by Hurlbut; it provides a structured, script-based description with a focus on interaction like most other use case notations [53]. Written use cases have been found to be also effective for generating test suites [12] and for generating security policies [14].

Our results indicate that there is no significant superiority between both notation types per se. However, we observe that a presentation of a process in both notations often provides improved results, which is in line with cognitive load theory and multimedia learning theory. Furthermore, we find that familiarity with one notation improves understanding, while textual aptitude implies good

use case understanding.

We structure the paper following the suggestions of Wohlin *et al* [76]. Section 2 explains the particular research question that we address. We refer to established cognitive theories for deriving a set of hypotheses about the impact of the type of notation on understanding performance. In Section 3, we discuss the appropriate research methods for our evaluation. We describe our research design, the demographics of participants, the instruments that the participants worked through, and the procedure the participants followed. We also describe how we controlled the experiments to enhance the reliability of the results. In Section 4, we present our experimental results and the statistical tests we applied. Section 5 discusses our findings in the light of related work, before Section 6 concludes the paper.

2. Background

In this section, we discuss the background of our research on understandability of notations for capturing processes. Section 2.1 summarizes findings on comparing notations in software engineering. Section 2.2 investigates theories, which are deemed relevant to the understanding of the cognitive phenomena contributing to superiority of particular notations. We conclude this discussion in Section 2.3 with a set of hypotheses.

2.1. Textual versus Graphical Notations

There is a long record of comparisons between competing notations in software engineering, summarized for instance in [17, 66]. Prior work comparing process modelling notations can be roughly grouped into two categories: (1) graphical notation comparison and (2) textual versus graphical notation comparison, with the former being the more prominent group.

Graphical notations for process modelling have been compared from different perspectives. These comparisons are based, for instance, on the workflow patterns [2], representational theory [58], or perceptual characteristics of the symbol set [15]. This stream of research reveals relative strengths and weaknesses of notations such as Petri nets, Event-driven Process Chains (EPCs), UML Activity Diagrams, YAWL, and Business Process modelling Notation (BPMN). All these languages are essentially flow charts. They represent a business process as a particular kind of graph with nodes capturing the activities and arcs defining the control flow. It is interesting to note that empirical work in this area has found only marginal difference in terms of comprehension performance. While the experiment by Sarshar and Loos [63] identifies a slight tendency towards better results with EPCs than with Petri nets, a more recent experiment by Recker and Dreiling does not find improved performance when using a particular notation like EPCs and BPMN [57]. On the other hand, it has been shown that flow charts appear to be superior in

conveying a general overview of a process in contrast to static diagram types such as class diagrams [4].

Prior research on textual versus graphical notations has discussed their relative strengths and weaknesses essentially based on two arguments. On the one hand, it is assumed that diagrams and graphical notations are easy for humans to understand. A seminal argument for the benefits of graphical over textual notation was given by Larkin and Simon [31]: While text is limited to a linear order, graphical diagram allow for a more efficient information processing due to the spatial arrangement of different elements on a modelling canvas. On the other hand, Siau points to the fact that symbols of a graphical notations need to be learnt by human readers in order to be understood [65]. Therefore, training is required before the benefits of a graphical notation can materialize. This is supported by findings on considerable error rates in graphical process models [40, 24, 37]. On the other hand, Moher *et al* [44] looked at several ways to express program structures in text and in diagrams (Petri Nets), and state “for our tasks, graphics were no better than text, and in several cases were considerably worse”. Altogether, current research does not provide a clear picture about the relative importance of each argument.

In this paper we focus on one graphical notation, namely BPMN [47], and one textual, which is a dialect of Cockburn’s written use case notation [3, 8]. Both notations are sufficiently expressive to describe reasonably sophisticated business processes, and are closely related to other process-based and text-based notations in this domain [53, 58, 75, 38].

- The Business Process Modelling Notation (BPMN) depicts the flow of a business process as a graph. Activities are the major type of nodes in such a graph (captured as rounded boxes) while the arcs define temporal and logical order. Activities are usually annotated with short text labels following a Verb-Object style, as for instance “place order” [42]. There are also diamond-shaped routing elements for describing decisions based on certain conditions or parallel execution. Actors are represented as so-called swimlanes, in which activities can be placed. Figure 2 uses BPMN to describe a business process.
- Written use cases (UC) are widely employed for requirements elicitation, but also for capturing business processes [48]. An activity is described on a textual way following a restricted grammatical as Subject-Verb-Object [8, Page 90]. A scenario is a partial specification consisting of partially ordered activities [8, page 33]. An extension is a scenario that is invoked under a stated condition [8, page 99]. A use case is a set of scenarios including a main success scenario and some alternative scenarios, each with its extensions [8, page 106]. Figure 1 applies the use case notation to describe a business process.

Use Case #2 Detail analysis and design
 Use case Scope: High level
 Trigger: Solution Concept Document (SCD) is completed and approved
 Primary Actor: Project Manager who has to assess the project cost in detail
 Actors: Project team, Technical Lead

Main success scenario

- 1: General Manager nominates a Project Manager. Order = 1
- 2: Project Manager supplements the solution concept document with detail, writing a Project Scope Document (PSD). Order = 2
- 3: Project Manager assembles the project team, including direct reports, customers, suppliers and auditors. Order = 3
- 4: Project team reviews Project Concept Document. Order = 4
- 5: Technical Lead writes High Level Design (HLD) evaluating several design avenues. Order = 5
- 6: Technical Lead writes an assessment of technology and methods selecting one of the design alternatives. Order = 6
- 7: Technical lead conducts vendor selection. Order = 7
- 8: Technical lead writes a Low Level Design (LLD) (Solution Design, Support impact, Risk assessment). Order = 8
- 9: Project manager writes Detailed Project Plan using input from the LLD (Time Line and Resources in Microsoft Project, Cash flow, Risk Management Plan). Order = 9

Extensions:

- 2a: The project manager finds the Solution Concept Document unrealistic
 - 2a1: The project ends
- 4a: The project team finds errors in the Project Concept Document
 - 4a1: The use case starts at action step 2
- 8a: While writing the LLD the technical lead finds errors in the HLD.
 - 8a1: The technical use case continues at action step 5.

Figure 1: Extended written use case example. The participant hand books included six models of similar complexity. A supplementary technical report includes the instruments we used in our experiment [49].

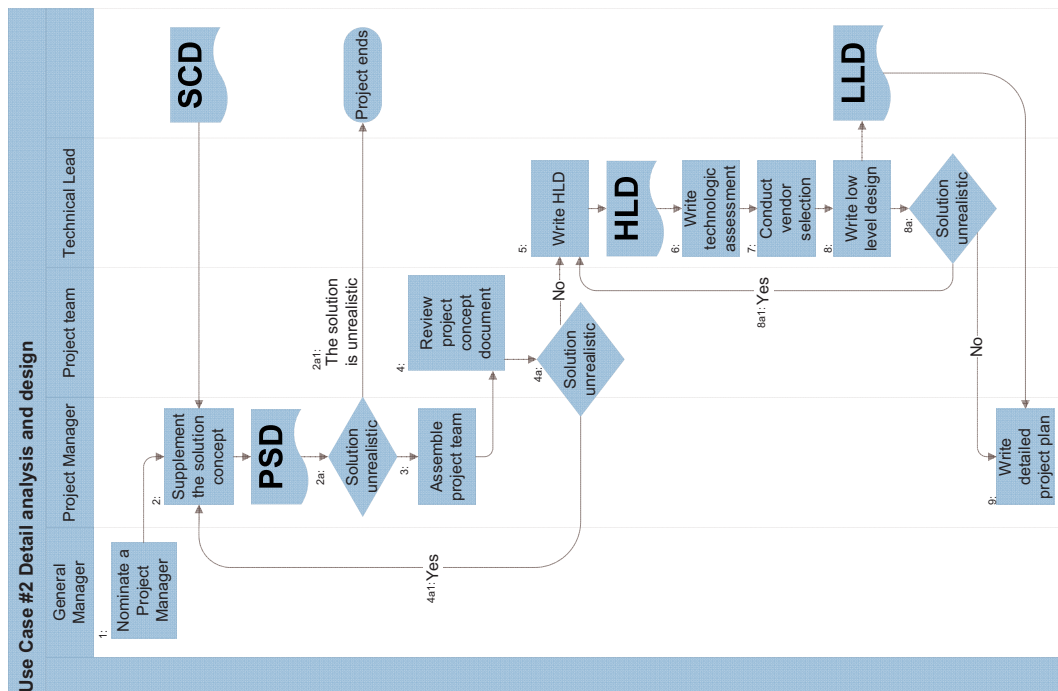


Figure 2: BPMN Example. The participant hand books included six models of similar complexity. A supplementary technical report includes the instruments we used in our experiment [49].

By focusing on two prominent and typical representatives of graphical and textual process modelling notations, we aim to contribute a general understanding of the relative strengths and weaknesses of these two notation types.

2.2. Cognitive Theories on Notations

For disentangling partially contradicting insights on graphical versus textual notations, it is instructive to consult relevant cognitive theories. In general, cognitive load theory suggests that the burden for a person to read a diagram should be low in order to facilitate an efficient cognitive processing [69]. The observation builds on characteristics of the human brain. In particular, its working memory is responsible for processing sensual information, e.g. from reading diagrams, with a restriction to about 30 seconds of storage duration and only a few matters (roughly seven). Once this number of seven items is crossed, humans lose track of the overall matter. Against this general background, there are theories emphasizing strengths of graphical notations, of textual notations, and of their joint usage. Finally, there are also insights that suggest personal characteristics to be important.

Several works identify a *strength of graphical notations* in terms of their efficient information processing. The essential advantage stems from computational offloading, i.e. processing is done already by the perceptual system, such that scarce resources of the cognitive system are spared [31, 45]. This visual processing is also much faster than cognitive processing. The physics of notations [45] emphasize some additional requirements for a notation to be visually effective. Among others, a notation should provide the full spectrum of retinal variables or a good perceptual discrimination between its symbols [45]. It is a *strength of textual notations* that they can be readily understood. In contrast, graphical notations require readers to have learnt the semantics of its symbols [65]. This can lead to significant reading effort for non-experts. For this reason, formal models are often translated into textual descriptions by system analysts, such that people working in a particular domain of concern can effectively validate them [16].

There are also good arguments for *using both notations* at the same time. Green [22] observed that while different notations can achieve identical ends, where the information structure they use is different, they facilitate different cognitive processes. In our case, while the notations can probably present the same information, the textual one may be better at managing a multitude of exceptions and the graphical notation may be better at managing a multitude of nested loops. As each notation highlights some types of information while obscuring other types, each notation may facilitate some tasks while making others harder. Therefore, the notations may not be absolutely good, but good only in relation to certain tasks. If so, we expect that presenting the same information using two notations would increase comprehension. This is in line with dual coding theory [50] and multimedia learn-

ing theory [36], which both suggest presenting text and diagrams together to reinforce a message.

Finally, it may also be expected that personal features of different people have an impact on understanding performance. Vessey [72] differentiates between a problem representation using a notation and its mental representation. Reading is thus a transformation from one representation to another. A good fit of a notation to a problem-solving style would simplify the reading process by requiring less transformation. Ideally, there should be no transformation from the notation based representation to the mental representation. Hence, a good cognitive fit of a notation to a thinking style would lead to effective and efficient problem-solving process. We would thus expect to see a difference between the understandings of the two notations depending on the thinking styles of the readers. In our case, since we are comparing a graphical notation with a textual one, we explore whether it mattered if a reader had a preference for textual or graphical information in other contexts. We also consider one's experience with a similar notation, as a possible predictor of the effective use of a notation.

2.3. Research Hypotheses

Against this background, it is the aim of our study to investigate these conflicting views of the understandability of graphical and textual notations, in a particular domain and with particular notations of each sort. We evaluate whether each notation does convey useful information to readers, and we also compare the extent of information gain by readers from the two notations. Thus, we first test hypotheses which claim that the textual notation is effective, and that the graphical notation is effective in conveying domain information.

H_1 : Reading a process model in written use case notation increases domain understanding.

H_2 : Reading a process model in BPMN notation increases domain understanding.

We next investigate two claims about the comparison between the notations, which are in fact mutually contradictory. These go to the heart of the debate about the suitable choice of a notation. They relate to cognitive research on comprehension efficiency and the ease of understanding of a notation's symbols.

H_3 : Reading a process model in written use case notation increases domain understanding more than reading a model in BPMN notation.

H_4 : Reading a process model in BPMN notation increases domain understanding more than reading a model in written use case notation.

Finally, we hypothesize that the effect of a joint usage of both notations provides a better understanding than when each is used alone. This assumption is supported by dual coding theory and multimedia learning theory.

H_5 : Reading a process model in written use case notation followed by a corresponding model in BPMN increases domain understanding more than only reading a model in written use case notation.

H_6 : Reading a process model in BPMN notation followed by a corresponding model in written use cases increases domain understanding more than only reading a model in BPMN notation.

Furthermore, we aim to investigate personal characteristics regarding their impact on understanding. In particular, we will consider the familiarity with a particular type of notation and textual aptitude. Cognitive fit theory suggests that both should correlate with a better performance of the corresponding notation types.

3. Experiment Planning and Operation

We use an experimental approach to investigate the hypotheses identified above. In Section 3.1, we discuss our research design. Section 3.2 describes the sample of participants. Section 3.3 details the instruments and the experimental procedure. Section 3.4 explains the control we imposed to guarantee validity of the experimental results.

3.1. Research Design

We utilize an experimental design, as it is frequently used in notational research [66, 17]. Our research design is characterized as a between-grammar study [17] in which we focus on reading of instruments [5]. We consider a between-subjects experiment design where each subject is assigned a different condition [13, Page 331]. For further control, we randomly assign participants. Our experiment is designed to take place in an off-line setting, i.e. it is not located within the bounds of a real project [76].

For the measurement of understanding, Gemino *et al* distinguish between “comprehension”, which refers to what the reader can answer about particular elements of the notation, and “domain understanding”, which is shown by problem-solving questions requiring significant additional cognitive processing [18]. To compare the effectiveness of artifacts in the usual way, researchers ask participant to read an instrument and then measures participants’ accuracy of answers to questions about the domain described in the instrument. A confounding aspect for this measurement might be the different levels of initial domain knowledge participants bring with them. Experiments sometimes implicitly assume homogeneity of initial domain knowledge within a well-known domain (*e.g.* restaurant [27], ballistic trajectory [23], elevator [25]) or else they generate synthetic artifacts, thus removing any domain specific information from the artifacts (*e.g.* [41, 73]). We have not witnessed an experiment comparing design notations that measures initial domain knowledge explicitly, rather, we have witnessed some experiments that

used self-assessment [57]. In contrast, in the medical area it is a common practise to compare a treatment against the use of a *placebo*, a substance with no therapeutic effect used as a control in testing new drugs [9, Accessed 2-June-2010].

Accordingly, we conduct a between-grammar study that measures understanding performance of a domain based on information conveyed via notational representation. We use both a graphical and a textual representation as a treatment. We vary the order of the two treatments randomly, and assign it to participants in a double blind way. We use both between-subjects and within-subjects comparisons. The multiple subjects are students, the multiple objects are taken from toy problems, and the experiment occurs off-line.

3.2. Participants in the Experiment

The common method for assessing understandability of a notation is to assign understanding tasks to people, and measure how well they grasp a domain by letting them answer questions [41, 42]. The result is the sum of three components: initial domain knowledge, contribution from the document, and chance. We refer to this total measurement as *absolute understandability*. We measure understandability by concentrating on the contribution, which is the component that the artifact creator controls. We refer to this measurement as *relative understandability*. Our experimental procedure aims at reducing chance, then gauging participants’ initial domain knowledge, and then assessing the absolute understandability. We arrive at relative understandability of the artifact by subtracting the initial domain knowledge from the absolute understandability.

196 participants, all university students, were drawn from three universities. 129 participants were post-graduate industrial engineering students from Eindhoven University of Technology, The Netherlands (TU/e). 26 participants were advanced business process management and enterprise systems post-graduate students from Humboldt-Universität zu Berlin, Germany (HU). They were encouraged to take part as the experiment was relevant to their studies. Both types of participants can be considered as proxies for business analysts (BAs), since they have received explicit training in business process modelling and flow-chart notations. Post-graduate students (like these) have been previously found to be adequate proxies for analysts with low to medium expertise levels [18, 51, 60].

The remaining 41 participants were students following various courses in the University of Sydney, Australia (USYD). They were recruited by advertisements on noticeboards in cafeterias, and paid AU\$20.00 for their effort. Participants from USYD are considered as proxies for business users (BUs) without training in flow-charting. They come from a broad range of disciplines, and are likely to act in that role a few years after joining the workforce. Note that the preliminary knowledge of business user participants is not relevant, as will become clear from the

discussion of the controls that have been applied in this experiment.

3.3. Instruments and Procedure

We asked participants to follow through a workbook. A supplementary technical report includes a sample instrument we used in our experiment [49]. The preamble to the workbooks was a disclosure statement, which was followed by a privacy statement. At the core of the workbooks resides a description of a certain business issue. The workbook also includes a placebo, a solution to the business issue, three identical questionnaires, a preferences survey, and a demographic survey. The length of the workbook is 39 printed A4 pages.

Being descriptions of a toy problem, the artifacts were shorter, poorer in red herrings, less ambiguous, and more consistent than real life artifacts. The workbook articulated the genuine solution to the business issue twice, once employing eight A4 size BPMN diagrams similar in complexity to that of the diagram in Figure 2, and once employing eight written use cases of similar complexity to that of the text in Figure 1. Each questionnaire had six multiple choice questions about the genuine solution to the business issue.

In the workbook, we asked the participants to complete the following procedure:

1. Read a disclosure statement explaining the experiment’s goals, the tasks, and the participants’ privacy.
2. Read, and optionally sign, a consent form. The participant could opt out at any stage.
3. Read description of a business issue that a project office in a hypothetical financial services company faces.
4. Read the placebo that describes the financial services that the hypothetical company offers. It includes no information related to the business issue that the project office faces.
5. Answer a multiple-choice questionnaire asking about factual matters concerning business processes that solve the business issue the project office faces, a solution that was not presented yet to the participants.
6. Read an artifact that describes the business processes in one notation.
7. Answer the same questionnaire for the second time.
8. Read a second artifact, presenting the same business processes in the other notation.
9. Answer the same questionnaire for the third time.
10. Fill in an attitude survey on personal preferences.
11. Fill in a demographic survey.

In order to compare the impact of different notations, we used two workbook types; in one type we used one notation in step 6 and the other notation in step 8; the other workbook type reversed the order in which the notations were given. We refer to one workbook type (and to the condition of participants who receive it) as “BPMN first”; the other condition is “written use cases first”.

3.4. Control

To arrive at reliable results, from which we will be able to generalize beyond the narrow scope of the experiment, we put in place several controls.

To neutralize initial domain knowledge, we subtracted the placebo score from both written use cases and BPMN scores, and arrived at the contributions of the two notations. In step 5, to convince the participants to answer questions after only seeing the placebo, we stated that “*the philosophy of this design notation is that one must understand the products of a company to understand its processes.*” We thus used the questionnaire presented in step 5 to measure the initial domain knowledge of the participants. In step 7 we measure the absolute understandability of the first notation, and in step 9 we measure the absolute understandability of a presentation through both notations in sequence. Later we refer to the results of the questionnaire presented in step 5 as *Placebo*, to the results of the questionnaire presented at step 7 as *QSet1* and to the results of the questionnaire presented in step 9 as *QSet2*. We define Primary Contribution as

$$PrimaryContribution = QSet1 - Placebo \quad (1)$$

and Secondary Contribution

$$SecondaryContribution = QSet2 - QSet1 \quad (2)$$

When presenting two different artifacts to participants, whether they are within-grammar or between-grammar, the artifacts should include equivalent information which Larkin *et al* define as “*Two representations are informationally equivalent if all of the information in the one is also inferable from the other, and vice versa. Each could be constructed from the information in the other*” [31]. Larkin *et al* also speak about computationally equivalent representations, but the criteria to evaluate that equivalence are subjective [17, 65]. To ensure equivalence between the BPMN and the written use case artifacts, we reconciled the designs with each other, ensuring that the use case set and the BPMN diagrams were (i) logically identical, (ii) included the same information, and (iii) included the same amount of information.

To neutralize allocation bias, we applied double blind sampling. We randomly sorted the workbooks and placed these in sealed unmarked envelopes, thus prevented unintentional bias in the allocation of participants to groups according to perceived verbal or graphical aptitude.

To ensure the correct interpretation of participants’ answers, the questions concentrated on the knowledge domain, the lowest level within Bloom’s cognitive section [30]. We asked questions such as: “*What happens if stakeholders change the project scope?*” or “*What condition determines when a build is reiterated?*”.

To ensure anonymity, participants were asked not to write their names on the workbooks and to remove the disclosure statement and the signed privacy statement from workbooks before commencing the experiment. To further

enhance the perceived anonymity of the participants, we added “*I do not wish to answer this question*” options to each question in the preferences and demographics part.

To allow the participants to fully articulate their understanding of the solution, or even criticise it, we included the following statements “*I do not know*”, “*The workbook does not supply information needed to answer the question*” and “*None of the above*” option to each question in the questionnaire. We randomised the choice sequence among the concrete answers. Finally, to ensure that participants answered questionnaires one by one, immediately after reading the appropriate instrument, we asked participants not to read ahead.

4. Data Analysis

Our main, randomly controlled, *Independent variable* was the order of the treatments, namely: (i) BPMN first and written use cases (abbreviated as “UC”) second, and (ii) written use cases first and BPMN second. Another aspect that varied was the group to which each subject belonged (proxies for BAs, or proxies for BUs); however we regard this not as an independent variable within one experiment, but rather, we consider a hypothesis on each group of subjects, and ask for its validity on both groups.

Our *dependent variables* were the *primary* and *secondary* contributions to the understandability. Recall that primary contribution is the score on the second questionnaire *minus* the first questionnaire (the Placebo). Similarly, secondary contribution is the change between the second and third questionnaire (due to seeing the second notation).

We use *R* and *Stata* [54, 68] for statistical analyses.

4.1. Descriptive Statistics

The descriptive statistics from our experiments are summarized in Table 1 and in Figures 3 and 4, with appropriate rounding. Details are available in full in a supplementary technical report [49].

As seen in Figure 3 the mean initial contribution of written use cases was reasonably consistent among the cohorts, ranging from 0.8 at USYD to 1.2 at HU. The initial contribution of BPMN varied widely (from 0.4 at USYD to 1.4 at TU/e), and it was bigger than the initial contribution of written use case for the participants from TU/e, but lower than written use cases at USYD. While the secondary contribution of BPMN to TU/e was modestly positive, and big at USYD, the remaining secondary contributions were small or even negative. An explanation that may be offered for this effect is that the participants had a more difficult time remembering the initial diagrams after being confronted with the second artifact. If we consider the combined impact of primary and secondary contribution, we see that presenting written use cases followed by BPMN gave consistent good results (from 1.2 to 1.7), while the reverse order had a wide variation (from 0.5 to 1.3).

Table 1: Results from the three universities.

University	First artifact	Score	\bar{x}	s	n
TU/e	BPMN	Placebo	1.96	1.09	74
		QSet1	3.41	1.32	
		QSet2	3.23	1.37	
	UC	Placebo	1.78	1.10	55
		QSet1	2.73	1.38	
		QSet2	3.16	1.36	
USYD	BPMN	Placebo	2.05	1.22	19
		QSet1	2.47	1.39	
		QSet2	2.53	1.07	
	UC	Placebo	1.86	0.77	22
		QSet1	2.68	1.09	
		QSet2	3.55	0.86	
HU	BPMN	Placebo	1.77	0.73	13
		QSet1	2.85	1.07	
		QSet2	2.54	1.27	
	UC	Placebo	1.62	1.12	13
		QSet1	2.77	1.54	
		QSet2	2.85	1.77	

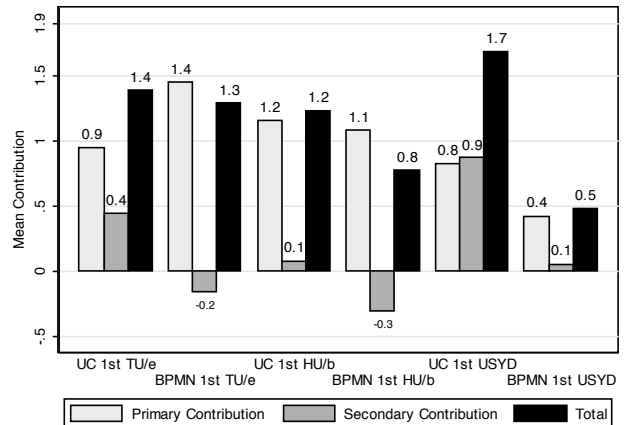


Figure 3: Mean Contributions – All samples

4.2. Discussion of the Data

While the theoretical range of contributions was -6 to 6 , the range of the contributions we measured was between -3 and 5 (a negative contribution implied that the instrument confused the participant). For ten of the twelve tests we performed the contributions were positive. The two exceptions were the TU/e and HU participants who received BPMN first. When subsequently presented with written use cases, their test results went down (see Figure 3 which compares the mean relative 1st contribution of each notation and of both notation to our six participant groups). In all the tests the initial domain knowledge (placebo) was bigger than the contribution of any design.

Our findings suggest that the data are distributed normally. The Shapiro-Wilk W Test for Normal Data for $N < 5000$ found that the test results are consistent with

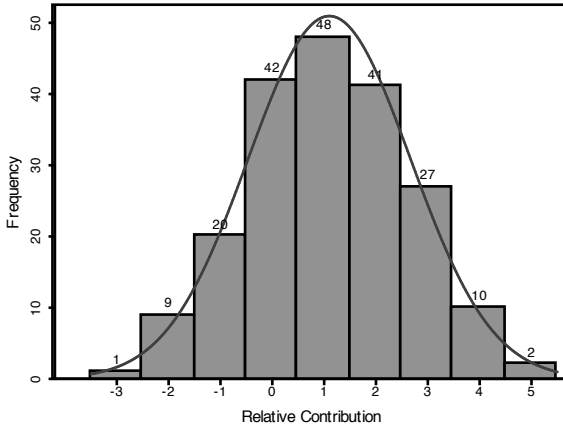


Figure 4: Distribution of the primary contribution – all participants – all notations

normality:

- $Placebo(Prob > z = 0.7552)$
- $QSet1(Prob > z = 0.9958)$ and
- $QSet2(Prob > z = 0.9982)$.

4.3. Hypothesis Testing

To compare the two notations, we performed a range of common statistical tests on hypotheses 1 to 6. Table 2 shows the p-values from the one-sided Wilcoxon tests, for the two groups of participants: students at TU/e and HU, as proxies for business analysts (BAs) and students at USYD, as proxies for business users (BUs). We use *italics* for p-values below 0.05. We also calculated other tests, such as t-tests; the details can be found in [49] but the significant conclusions are the same.

There is support for H_1 (written use-case notation provides an increment in understanding) and H_5 (written use-case followed by BPMN provides better understanding than written use-case alone) at the statistically high significant 0.01 level, among both groups. If we consider only the business analysts proxies, we see strong support for H_2 (BPMN does contribute) and also support that is significant at 0.1 level for H_4 (BPMN contributes more than written use-case); however, neither of these hypotheses seems well-supported for the business user proxies. It is important for effective communication that a notation be read correctly by diverse groups, including both business analysts and business users, so we see the data indicating that BPMN on its own is not a sufficient way to present process designs, despite its success among those with training in process models. Instead, our experiment offers reasons to provide a written presentation first, and then follow it with a graphical equivalent; this gave maximal scores among all the communities of readers.

Table 2: P Values for one-sided Wilcoxon sign-rank tests

Hypotheses	BA	BU	Subjects	Data set 1	Data set 2
H_i	Proxies	Proxies	Test	Contribution	Contribution
H_1	<i>0.0000</i>	<i>0.0054</i>	Within	Primary (UC)	0
H_2	<i>0.0000</i>	0.1540	Within	Primary (BPMN)	0
H_3	0.9462	0.2352	Between	Primary (UC)	Primary (BPMN)
H_4	0.0542	0.7729	Between	Primary (BPMN)	Primary (UC)
H_5	<i>0.0066</i>	<i>0.0003</i>	Within	Secondary (BPMN)	0
H_6	0.9114	0.4521	Within	Secondary (UC)	0

4.4. Further Analyses

While searching for attributes that would help predict understandability of a notation beyond the hypotheses tested above, we asked participants to rate their comfort and experience with BPMN and written use cases using scales of one to five for each notation. We observed four OLS (ordinary least squares) linear regressions models as described below. The model-estimates we provide below are pairs of coefficients, one for the high scale and one for the low. We qualify each coefficient, in brackets, by a standard error, and a t-value. The coefficient reflects the weight of the variable and the impact of one unit change in the variable on the primary contribution of the notation. The standard error shows the variability in the weight calculated. The t-value, which is the coefficient divided by the standard error, indicates significance for a two tailed test. If the absolute value of the t-value is greater than 1.96, we have confidence that random chance is not likely to lead to the observed impact, so we say that the relevant attribute makes a significant difference in performance.

When asked if they were comfortable with flow charts in general, 82 participants responded that they strongly agree and one participant responded with strong disagreement. The remaining participants reported preferences somewhere in the middle. Participants at the higher end of the scale did not perform differently when compared to those at the lower end, with model estimates of -0.2258 (0.1507; -1.4980) for comfort, and 0.0642 (0.1576; 0.4075) for lack of comfort.

When asked if they were comfortable with written use cases, 73 participants responded that they strongly agree and 16 participants strongly disagreed. The remaining participants reported preferences somewhere in the middle. Participants at the higher end of the scale did not perform differently when compared to those at the lower end, with model estimates of -0.0823 (0.0986; -0.8350) for comfort and 0.0701 (0.1027; 0.6830) for lack of comfort.

When asked if they often worked with flow charts, 30 participants strongly agreed and six participant responded with strong disagreement. The remaining participants reported preferences somewhere in the middle. Participants at the higher end of the scale performed significantly better, with model estimates 0.2669 (0.1558; 2.3104). The performance of participants at the lower end varied with

estimates -0.0865 (0.1351; -0.6404).

When asked if they often worked with written use cases, 36 participants responded that they strongly agree and 20 participants responded with strong disagreement. The remaining participants reported preferences somewhere in the middle. Participants at the higher end of the scale performed significantly better at the 0.1 level; the model estimates are 1.1875 (0.6426; 1.8479). The performance of participants at the lower end of the scale varied, with model estimates 0.1714 (0.3569; 0.4803).

We found textual aptitude to be a statistically significant predictor. When asked how many fiction books they had read in the past 12 month, 73 participants reported none, 52 reported one to three, 26 reported four to six, 13 reported seven to ten, and 22 reported more than ten. *Our findings suggest a strong relationship between readership and understanding of the written use case notation.* For the participants who first received written use cases as opposed to BPMN, the number of books they read were clustered via a Wards Hierarchical clustering routine, which revealed three distinct clusters of readership intensity. The first cluster comprised low readership, the third cluster comprised high readership levels. The primary contribution of the written use cases was regressed over readership intensity. Low readership significantly predicted a negative effect on the primary contribution with written use cases: -0.2370 (0.1125; -2.1070) and high readership predicted a significant positive effect: 0.4712 (0.1918; 2.4560). We also investigated this factor among the participants who received BPMN first: high readership predicted a significant negative effect on primary contribution from the BPMN artifact: -0.3911 (0.0867; -4.5092), but low readership was not a statistically significant predictor for success with BPMN: -0.1250 (0.1814; -0.6890).

4.5. Interpretation of results

Our analysis suggest that *participants from all groups can understand a design presented in written use cases, and business analysts benefit from flow chart models.* Our results support H_1 . Proxies for business users and for business analysts showed statistically significant increases in their understanding of designs after reading a written use case presentation, when compared to their understanding using only background knowledge of the domain. In contrast, only proxies for business analysts showed statistically significant increases in their understanding of designs after reading a BPMN presentation when compared to their understanding from background knowledge of the domain.

Our findings show that participants from all groups who first read written use cases benefited further from the BPMN set. This was not observed in the reverse order. Our results support H_5 : proxies for business users and proxies for business analysts showed statistically significant increases in their understanding of a business problem from reading BPMN model following the delivery of written use cases. In contrast, the delivery of written use

cases following the delivery of BPMN increased comprehension for business users but decreased it in business analysts. Dix *et al* state that formal specification should be accompanied by extensive commentary and a parallel textual description [13, Page 596] and our results support the complementary nature of both notation types.

The drop from TU and HU students in understanding after first reading a BPMN model and then reading a use case can be discussed in the light of the following two options. First, the reason for this drop could have been a fatigue problem with participants from Europe. This explanation is not likely though, because such a fatigue the effect should be visible also for the other group that got the use case first. Second, there is research that demonstrates the strength of a graphic to enhance text understanding [20, 36], maybe this does not hold the other way round. This proposition indeed seems reasonable under the condition that the textual notation was unfamiliar to the readers. Curtis *et al.* observed in [10] that individual difference account for about two third of performance difference while notational choice was of secondary importance. Also Roth and Bowen note that even experts have significant problems with comprehending unfamiliar notations [61]. Indeed, both the TU and the HU students were highly familiar with BPMN, but not with the use case notation.

The recruitment procedures we applied did not generate a random representative sample of the business analysts and business users in the industry. Nonetheless, there are some considerations that support the results to have external validity. Business users in a workplace had been students in a wide variety of fields, only a few years earlier. Both share the characteristic that they typically do neither have training in reading formal models nor modelling skills in general. The main difference that we would expect between students and business users is the level of initial domain knowledge, and we explicitly controlled for the effects of this in our analysis.

Similarly, post-graduate students who study industrial engineering or business process modelling have been shown previously to be valid proxies for business analysts with low or median expertise in the industry [18, 51, 60]. These groups are both familiar with the concepts of formal models. Indeed, these cohorts of students demonstrated a good grasp of the particular notations in our experiment, with higher primary contribution from each notation than the generalist students from Sydney.

Our sense that the conclusions are robust is also aided by the big size of the populations, the low p-values we found for some of the tests, and the strict control we exercised over the experimental procedure. We suggest that the amount of fiction reading is predictive for the level of understanding of written use cases. Other aspects of graphical or textual aptitude, experience or preferences do not seem particularly informative.

5. Related Work

In general related work on the research presented in this paper can be organized in two groups: first, work on comparing business process modelling notations, and second, work on comparing textual and graphical notations for programming.

There is extensive work that compares different business process modelling notations from a conceptual point of view, e.g. using the workflow patterns [2], representational analysis [58], or perceptual characteristics of the symbol set [15]. Here, we focus on empirical work in the context of process modelling; in the more general area of conceptual modelling empirical work is summarized in [18].

Agarwal *et al* [4] were among the first to explore the relative strengths and weaknesses of a business process modelling notation in comparison to an object-oriented representation. Their experiment revealed that comprehension was good in all groups, but for hard questions participants receiving a process-oriented design did better. The experiment by Sarshar and Loos [63] investigates the relative understandability of Petri nets and Event-driven Process Chains (EPCs). Their findings suggest that EPCs might be slightly better. On the downside, there is no theoretical principle identified that would be responsible for this result. In contrast, the experiment by Recker and Dreiling does not find any performance difference when using a particular notation like EPCs and BPMN [57]. More recently though, Recker notes that differences in ontological expressiveness of EPCs and BPMN have an impact on the intention of a modeler to continuously use a notation [55]. It has to be noted for this set of experiments that several model-related factors were controlled. In other works it has been shown that whether the information in the model is well organized in terms of labeling [42], secondary notation [59], iconic symbol design [67, 45], or structuredness [33] has an important influence on understanding. In the same vein, a variation in complexity of the process model in terms of size and other metrics [7, 24, 43, 38] and individual differences [60] results in different levels of understanding. Our results confirm this stream of research by emphasizing the importance of individual difference in terms of notation familiarity and cognitive fit, and highlight the relevance of textual aptitude for text-based notations.

The relative strengths and weaknesses of textual versus flowcharting notations has been investigated in a series of studies. Whitley provides a corresponding overview [74]. The arguments brought forth by Larkin and Simon emphasize the strength of visual representations [31], which is partially supported by Mayer [35] and Roth [61]. The works by Green and Petre [52] and by Moher *et al* does not directly follow this reasoning [44]. Both studies emphasize the importance of secondary notation (like suitable layout) as a way of giving cues about the intension of a diagram. In this way, the understandability of a diagram becomes dependent upon the expertise of the modeler or model reading, as experts are more likely to understand

such cues correctly. In the study by Moher, where different Petri nets were compared to text, the nets showed consistently worse understanding than the corresponding textual representation [44]. Instead of showing superiority of one or the other type of notation, our research demonstrates the complementary character of text and graphics, which is in line with the studies of Glenberg, of Mayer, and of Kim [20, 36, 27]. Our findings on the importance of individual characteristics is partially supportive of work that stresses cognitive fit, e.g. [19, 72]. Yet, this support rather refers to a fit with mental structures of participants. Familiarity with notation is an important factor to understand the different patterns of performance. Also the importance of textual aptitude confirms the importance of individual difference as observed in text graphics comparisons [10, 60].

Understandability is much researched by the medical community where informed consent is mandated by law [64] and good communication reduces patient litigation against doctors [46]. Davis *et al* [11] compared the understandability of two consent forms using verbal interviews of 183 adults recruited from private and university oncology clinics and a low-income housing complex. In our terms, this experiment measured absolute understandability (they assumed that the interviewee population's initial domain knowledge was homogenous). The consent forms varied in degree of information, so there was not information equivalence.

6. Conclusion

In this paper, we investigated the strengths and weaknesses of graphical and textual notations from an empirical perspective. We conducted an experiment to compare the relative understandability of four alternative presentations: (i) a use case set on its own, (ii) a set of BPMN models on its own, (iii) a set of BPMN models followed by a use case set, and (iv) a use case set followed by a set of BPMN models.

In our experiment, we involved university students as proxies for two different types of communities that are likely to use such notations, i.e. business analysts and business users. The proxies for both communities well understood the written use cases, while our proxies for business analysts well understood the BPMN models. Among the alternatives we investigated, we found that presenting a business process twice, in the sequence of a written use case set followed by its equivalent set of BPMN diagrams, was the most effective way to build up comprehension of the process in question. Our findings suggest that this is the case regardless of the graphical or verbal aptitude, experience, and preferences of the individual participants. We did not reach conclusions about whether this outcome is due to general differences between text and diagrams, or is instead due to specific features of the chosen notations.

A possible explanation for the effective usage of use cases by both types of participants may be that generic

reading skills already form a sufficient basis for their usage, as suggested by the inspection of readership that we reported on. By contrast, an effective usage of BPMN models as a means of communication seems to build upon more specific training. Considering that a notation such as BPMN is explicitly positioned as to be readily used by and useful for all business users [47], it would probably be a mistake to assume that the skills to read such models are naturally acquired. Especially this insight may benefit those business users who will depend on graphical descriptions (of business processes in their daily work but like the proxies in our study) if they lack dedicated training in the usage of the notation.

In future research we aim to further investigate the effects observed in this study. In particular, it appears to be important to analyze to what extent prior knowledge of and expertise with a particular notation has an impact on one's performance. Therefore, we plan to conduct experiments with notation experts from practice and contrast their performance with first-year students. Furthermore, we are interested in gauging the impact of various types of training on business users' performance in making sense of graphical descriptions of business processes. Finally, it seems valuable to extend the range of notations under study, both textual and graphical, to determine whether their specific traits are of any impact on our findings.

Bibliography

- [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] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.
- [3] S. Adolph, P. Bramble, A. Cockburn, and A. Pols. *Patterns for Effective Use Cases*. Addison Wesley, 2003.
- [4] R. Agarwal, P. De, and AP Sinha. Comprehending object and process models: An empirical study. *IEEE Transactions on Software Engineering*, 25(4):541–556, 1999.
- [5] B. Albert and T. Tullis. *Measuring the user experience*. Morgan Kaufmann, 2008.
- [6] W. Bandara, G.G. Gable, and M. Rosemann. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, 14(4):347–360, 2005.
- [7] G. Canfora, F. García, M. Piattini, F. Ruiz, and C.A. Visaggio. A family of experiments to validate metrics for software process models. *Journal of Systems and Software*, 77(2):113–129, 2005.
- [8] A. Cockburn. *Writing Effective Use Cases*. Addison Wesley, 1999.
- [9] Various Contributors. *The Oxford English Dictionary*. 2nd ed, on line edition, June 1989.
- [10] B. Curtis, S.B. Sheppard, E. Kruesi-Bailey, J. Bailey, and D. Boehm-Davis. Experimental evaluation of software documentation formats. *Journal of Systems and Software*, 9(2):167–207, 1989.
- [11] T. C. Davis, R. F. Holcombe, H. J. Berkel, S. Pramanik, and S. G. Divers. Informed consent for clinical trials: a comparative study of standard versus simplified forms. *Journal of the National Cancer Institute*, 90(9), May 1998.
- [12] A.L.L. de Figueiredo, W.L. Andrade, and P.D.L. Machado. Generating interaction test cases for mobile phone systems from

- use case specifications. *ACM SIGSOFT Software Engineering Notes*, 31, November 2006.
- [13] A. Dix, L. Finlay, G. Abowd, and R. Beale. *Human Computer Interaction*. Pearson Education, 2004.
- [14] E. B. Fernandez and J. C. Hawkins. Determining role rights from use cases. In *RBAC '97: Proceedings of the second ACM workshop on Role-based access control*, pages 121–125, New York, NY, USA, 1997. ACM.
- [15] K. Figl, J. Mendling, M. Strembeck, and J. Recker. On the cognitive effectiveness of routing symbols in process modeling languages. In Witold Abramowicz and Robert Tolksdorf, editors, *Business Information Systems, 13th International Conference, BIS 2010, Berlin, Germany, May 3-5, 2010. Proceedings*, volume 47 of *Lecture Notes in Business Information Processing*, pages 230–241. Springer, 2010.
- [16] P.J.M. Frederiks and T.P. van der Weide. Information modeling: The process and the required competencies of its participants. *Data & Knowledge Engineering*, 58(1):4–20, 2006.
- [17] A. Gemino and Y. Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4):248–260, November 2004.
- [18] A. Gemino and Y. Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requir. Eng.*, 9(4):248–260, 2004.
- [19] D.J. Gilmore and T.R.G. Green. Comprehension and recall of miniature programs. *International Journal of Man-Machine Studies*, 21(1):31–48, 1984.
- [20] A.M. Glenberg and W.E. Langston. Comprehension of illustrated text: Pictures help to build mental models* 1. *Journal of memory and language*, 31(2):129–151, 1992.
- [21] B. Graaf and A. van Deursen. Visualization of domain-specific modelling languages using UML. In *Proc. ECBS'07: International Conference and Workshop on the Engineering of Computer Based Systems*, pages 586–595, 2007.
- [22] T. R. G. Green. Cognitive dimensions of notations. In *People and Computers V*, pages 443–460. University Press, 1989.
- [23] T.R.G. Green and M. Petre. Usability analysis of visual programming environments: A ‘cognitive dimensions’ framework. *Journal of Visual Languages & Computing*, 7:131–174, June 1996.
- [24] V. Gruhn and R. Laue. What business process modelers can learn from programmers. *Science of Computer Programming*, 65(1):4–13, 2007.
- [25] R. Guindon. Designing the design process: exploiting opportunistic thoughts. *Hum.-Comput. Interact.*, 5(2):305–344, 1990.
- [26] T. Jokela, N. Iivari, J. Matero, and M. Karukka. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In *Proc. CLIHC '03: Latin American conference on Human-computer interaction*, pages 53–60. ACM, 2003.
- [27] J. Kim, J. Hahn, and H. Hahn. How do we understand a system with (so) many diagrams? cognitive integration processes in diagrammatic reasoning. *Information Systems Research*, 11(3):284–303, 2000.
- [28] N. Kock, J. Verville, A. Danesh-Pajou, and D. DeLuca. Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. *Decision Support Systems*, 46(2):562–575, 2009.
- [29] S. Konrad and B. H. C. Cheng. Automated analysis of natural language properties for UML models. In *MoDELS'05 Satellite Events (LNCS 3844)*, pages 48–57, 2006.
- [30] D. R. Krathwohl. A Revision of Bloom’s Taxonomy: An Overview. *Theory Into Practice*, 41(4):212–218, 2002.
- [31] J. H. Larkin and H. A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987.
- [32] A. Lauder and S. Kent. Precise visual specification of design patterns. In E. Jul, editor, *Proc. ECOOP'98: 12th European Conference on Object-Oriented Programming*, pages 114–134, 1998.
- [33] Ralf Laue and J. Mendling. Structuredness and its significance for correctness of process models. *Inf. Syst. E-Business Management*, 8(3):287–307, 2010.
- [34] D. Leffingwell and D Widrig. *Managing software requirements: a unified approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000. ISBN 0-201-61593-2.
- [35] R.E. Mayer and J.K. Gallini. When is an illustration worth ten thousand words. *Journal of Educational Psychology*, 82(4):715–726, 1990.
- [36] Richard E. Mayer. *Multimedia Learning*. Cambridge University Press, Cambridge, Massachusetts, 2001.
- [37] J. Mendling. Empirical studies in process model verification. *T. Petri Nets and Other Models of Concurrency*, 5460:208–224.
- [38] J. Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.
- [39] J. Mendling. Event-driven process chains (epc). In W. M. P. van der Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, and C. Szyperski, editors, *Metrics for Process Models*, volume 6 of *Lecture Notes in Business Information Processing*, pages 17–57. Springer Berlin Heidelberg, 2009.
- [40] J. Mendling, M. Moser, G. Neumann, H.M.W. Verbeek, B.F. van Dongen, and W.M.P. van der Aalst. Faulty EPCs in the SAP Reference Model. In J.L. Fiadeiro S. Dustdar and A. Sheth, editors, *Proceedings of BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 451–457, Vienna, Austria, 2006. Springer-Verlag.
- [41] J. Mendling, H. A. Reijers, and J. Cardoso. What makes process models understandable? In G. Alonso, P. Dadam, and M. Rosenmann, editors, *Proc. BPM'07: Business Process Management (LNCS 4714)*, pages 48–63, 2007.
- [42] J. Mendling, H. A. Reijers, and J. Recker. Activity labeling in process modeling: Empirical insights and recommendations. *Inf. Syst.*, 35(4):467–482, 2010.
- [43] J. Mendling, H.M.W. Verbeek, B.F. van Dongen, W.M.P. van der Aalst, and G. Neumann. Detection and Prediction of Errors in EPCs of the SAP Reference Model. *Data & Knowledge Engineering*, 64(1):312–329, January 2008.
- [44] T. G. Moher, D. C. Mak, B. Blumenthal, and L. M. Leventhal. Comparing the comprehensibility of textual and graphical programs: the case of petri nets. In *Empirical Studies of Programmers: Fifth Workshop*, pages 137–161. Norwood, NJ: Ablex., 1993.
- [45] D. L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Software Eng.*, 35(6):756–779, 2009.
- [46] D. O’Connell and V.F. Keller. Communication: A risk management tool. *Journal of Clinical Outcomes Management*, 6(1):35–38, 1999.
- [47] OMG. *Business Process Model and Notation (BPMN) Version 2.0*, January 2011. <http://www.omg.org/spec/BPMN/2.0/>.
- [48] A. Ottensooser and A. Fekete. An enactment-engine based on use-cases. In G. Alonso, P. Dadam, and M. Rosemann, editors, *Proc. BPM'07: Business Process Management (LNCS 4714)*, pages 230–245, Heidelberg, 2007. Springer-Verlag.
- [49] A. Ottensooser and A. Fekete. Comparing readability of graphical and sentential process design notations – data analysis report. Technical report, School of Information Technologies, University of Sydney, Australia, 2010. TR-658, ISBN 978-1-74210-198-9.
- [50] Allan Paivio. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology*, 45(3):255–287, 1991.
- [51] J. Parsons and L. Cole. What do the pictures mean? guidelines for experimental evaluation of representation fidelity in diagrammatic conceptual modeling techniques. *Data & Knowledge Engineering*, 55(3):327342, 2005.
- [52] M. Petre and T.R.G. Green. Learning to read graphics: Some evidence that ‘seeing’ an information display is an acquired skill. *Journal of Visual Languages & Computing*, 4(1):55–70, 1993.
- [53] Hurlbut R. A survey of approaches for describing and formalizing use-cases. Technical Report 97-03, Department of Computer

- Science, Illinois, Institute of Technology, USA., 1997.
- [54] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.
- [55] J. Recker. Explaining usage of process modeling grammars: Comparing three theoretical models in the study of two grammars. *Information & Management*, 47(5-6):316–324, 2010.
- [56] J. Recker. Opportunities and constraints: the current struggle with bpmn. *Business Process Management Journal.*, 16(1):181–201, 2010.
- [57] J. Recker and 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 *18th Australasian Conference on Information Systems*, pages 356–366, 2007.
- [58] J. Recker, M. Rosemann, M. Indulska, and P. Green. Business process modeling: A comparative analysis. *Journal of the Association for Information Systems*, 10(4):333–363, 2009.
- [59] H.A. Reijers, T. Freytag, J. Mendling, and A. Eckleder. Syntax highlighting in business process models. *Decision Support Systems*, 2011 (doi:10.1016/j.dss.2010.12.013).
- [60] H.A. Reijers and J. Mendling. A study into the factors that influence the understandability of business process models. *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 2010. to appear.
- [61] W.M. Roth and G.M. Bowen. When are graphs worth ten thousand words? An expert-expert study. *Cognition and Instruction*, 21(4):429–473, 2003.
- [62] N. Russell, P. Wohed, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. chapter Pattern-Based Analysis of the Control-Flow Perspective of UML Activity Diagrams, pages 63–78. 2005.
- [63] K. Sarshar and P. Loos. Comparing the control-flow of epc and petri net from the end-user perspective. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, LNCS 3649, pages 434–439, 2005.
- [64] M.M. Sculz. From informed consent to patient choice: a new protected interest. *The Yale Law Journal*, 95(21), 1985.
- [65] K. Siau. Informational and computational equivalence in comparing information modeling methods. *Journal of Database Management*, 15(1):73–86, Jan.-Mar 2004.
- [66] K. Siau and M. Rossi. Evaluation techniques for systems analysis and design modelling methods – a review and comparative analysis. *Information Systems Journal*, 2008.
- [67] K. Siau and Y. Tian. A semiotic analysis of unified modeling language graphical notations. *Requirements Engineering*, 14(1):15–26, Feb 2009.
- [68] StataCorp Corporation. *Stata – Statistical Software: Release 7.0*. College Station, TX, 2001.
- [69] J. Sweller, J.J.G. Van Merriënboer, and F.G.W.C. Paas. Cognitive architecture and instructional design. *Educational psychology review*, 10(3):251–296, 1998.
- [70] R. N. Taylor and A. van der Hoek. Software design and architecture: The once and future focus of software engineering. In *FOSE '07: Future of Software Engineering*, pages 226–243, Washington, DC, USA, 2007. IEEE Computer Society.
- [71] S. Taylor. Extreme terseness: Some languages are more agile than others. In M. Marchesi and G. Succi, editors, *Proc. XP'03*, pages 334–336, 2003.
- [72] I. Vessey. Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decision Sciences*, 22(2):219–240, 1991.
- [73] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.
- [74] K.N. Whitley. Visual programming languages and the empirical evidence for and against. *Journal of Visual Languages and Computing*, 8(1):109–142, 1997.
- [75] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, and N. Russell. On the Suitability of BPMN for Business Process Modelling. In S. Dustdar, J. Fiadero, and A. Sheth, editors, *Proc. BPM'06: Business Process Management (LNCS 4102)*, pages 161–176, 2006.
- [76] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.