# The Impact of Change Task Type on Maintainability of Process Models

Matthias Weidlich[1], Stefan Zugal[2], Jakob Pinggera[2], Dirk Fahland[3], Barbara Weber[2], Hajo A. Reijers[4], and Jan Mendling[3]

[1] Hasso-Plattner-Institute, University of Potsdam, Germany
matthias.weidlich@hpi.uni-potsdam.de
[2] University of Innsbruck, Austria
stefan.zugal|jakob.pinggera|barbara.weber@uibk.ac.at
[3] Humboldt-Universität zu Berlin, Germany
fahland@informatik.hu-berlin.de|jan.mendling@wiwi.hu-berlin.de
[4] Eindhoven University of Technology, The Netherlands
h.a.reijers@tue.nl

**Abstract.** While process modeling has become important for documenting business operations and automating workflow execution, there are serious issues with efficiently and effectively creating and modifying process models. While prior research has mainly investigated process model comprehension, there is hardly any work on maintainability of process models. Cognitive research into software program comprehension has demonstrated that imperative programs are strong in conveying sequential information while obfuscating circumstantial information. This paper addresses the question whether these findings can be transferred to process model maintenance. In particular, it investigates whether it is easier to incorporate sequential change requirements in imperative process models compared to circumstantial change requirements. To address this question this paper presents results from a controlled experiment providing evidence that the type of change (sequential versus circumstantial) has an effect on the accuracy of process models. For performance indicators modeling speed, correctness, and cognitive load no statistically significant differences could be identified.

## 1 Introduction

The increasing use of business process models has sparked a discussion on usability and quality issues. Large companies use business process modeling as an instrument to document their operations typically resulting in several thousand process models, which are partially created by staff members with limited modeling expertise. Therefore, analyzing factors that influence the usability of process models is a promising approach for securing success of process modeling initiatives [2].

Prior research has mainly investigated process model comprehension as a prerequisite for usability. Among others, modeling expertise and process model complexity have been identified as factors of comprehension [15]. Yet, comprehension captures only a partial dimension of usability. Process models in nowadays process modeling initiatives are subject to frequent changes and a considerable amount of staff members are involved in updating process models. For this reason, investigating *process model maintainability* bears the potential to improve current process modeling practice.

Up until now, there is hardly any work on maintainability of process models beyond research into complexity metrics [3]. In this paper, we analyze in how far cognitive research into software program comprehension can be transferred to process model maintenance. Work on the cognitive dimensions framework has established a relativist view on usability [7, 9, 8]. In particular, it was demonstrated that imperative programs are strong in conveying sequential information while obfuscating circumstantial information. In this context, *sequential* information explains how input conditions lead to a certain outcome, and *circumstantial* information relates to the overall constraints that hold when that outcome is produced. We challenge this hypothesis for imperative process models in BPMN and test whether maintainability is influenced by the type of change requirement. Accordingly, we conduct an experiment that checks if sequential change requirements are easier to implement for a BPMN model than circumstantial change requirements. The results of this experiment foster research on maintainability factors of process models.

The remainder of the paper is structured as follows. Section 2 discusses the background of our research, namely sequential and circumstantial change requirements. Section 3 describes the setup for our experiment, which builds on a realistic modeling task taken from the disaster management domain. Section 4 covers the execution and the experiment's results. Finally, Section 5 discusses related work, followed by a conclusion.

## 2 Background

The central subject to maintainability considerations is the notion of a process change. A *process change* is the transformation of an initial process model $S$ to a new process model $S'$ by applying a set of change operations. A change operation modifies the initial process model by altering the set of activities and their order relations [12]. Typical change primitives are *add node*, *add edge*, *delete node*, or *delete edge* [21]. Figure 1 shows a BPMN process model from the domain of earthquake response, which is a simplified version of a process run by the "Task Force Earthquakes" of the German Research Center for Geosciences (GFZ). The main purpose of the task force is to coordinate the allocation of an interdisciplinary scientific-technical expert team after catastrophic earthquakes worldwide [6].

According to considerations on cognitive software program analysis, not all change requirements are equally difficult (cf., [7]). Here, we call a change requirement *sequential* if an activity has to be added, deleted, or moved directly before or behind another activity. For example, once arrived in the host country,

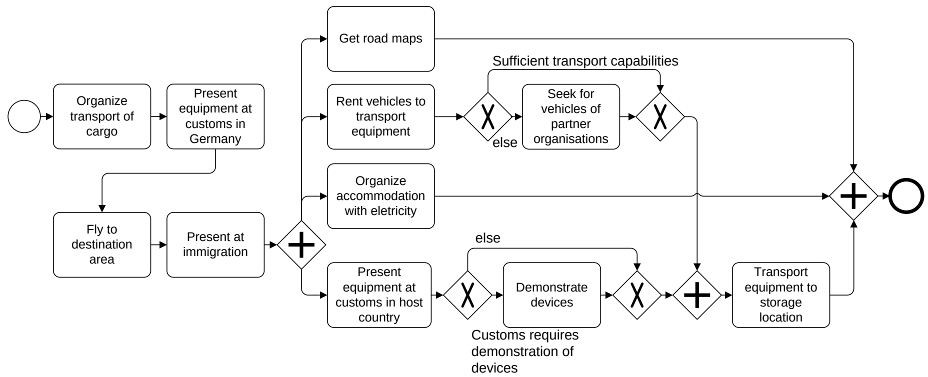


**Fig. 1.** BPMN Model for Transport of Equipment

the taskforce has to demonstrate the devices to customs (cf., Fig. 1). In contrast to the model of Fig. 1, customs might not clear the equipment which requires additional activities. A concrete change might be to insert an activity "Negotiate with customs" in the process after "Demonstration of devices." Such a sequential change requirement describes whether a pair of activities is in a specific structural or behavioral order relation. In contrast, a *circumstantial* change requirement involves adding or moving an activity such that a general behavioral constraint is satisfied. Such a constraint might be given in terms of temporal operators like always, eventually, until, and next time. As an example, consider a change requirement to execute "Demonstration of devices" eventually in every case. The region in the process model that needs to be changed cannot be deduced from the change requirement directly. Consequently, sequential changes tend to be rather *local* in the process model, whereas circumstantial changes tend to affect the process model *globally*. Two realistic change requirements are given in Appendix A.

How do these observations on process models relate to established theories? Adapting a software program to evolving needs involves both *sense-making tasks* (i.e., to determine which changes have to be made) and *action tasks* (i.e., to apply the respective changes to the program) [9]. We can discuss the problem of changing a process model in a similar vein. When a process designer is faced with a change requirement, they have to consider two things: 1) they need to determine which change operations have to be used to modify the process model; and 2) they have to apply the respective changes to the process model. Consequently, the effort needed to perform a particular process model change is on the one hand determined by the cognitive load to decide which changes have to be made to the model, which is a comprehension and sense-making task. On the other hand, the effort covers the number of edit operations required to conduct these changes, which is an action task.

In the cognitive dimensions framework, an important result–regarding sense making of information artifacts–founds on the difference between the tasks of looking for sequential and circumstantial information in a software program. Transferring this result to process models reads as follows: circumstantial changes are more difficult to perform on a flow chart diagram like BPMN [9]. Consequently, we would expect that process designers show a better performance in applying sequential change requirements. We challenge this hypothesis in an experimental setup.

## 3 Research Setup

In this section we describe the design of an experiment investigating the influence of different change types on the modeling performance.
**Subjects**: In our experiment, the subjects are 15 students in Software Engineering of a graduate course on Business Process Management at the Hasso Plattner Institute. Participation in the study was voluntary.
**Objects**: Object of our experiment is a process model along with two descriptions of a change that have to be applied to the model. The process model used in our experiment describes an actual process run by the "Task Force Earthquakes" of the German Research Center for Geosciences (GFZ) [6]. In particular, we used a model of the "Transport of Equipment" process similar to the one shown in Fig. 1 that specifies how the transport of scientific equipment from Germany to the disaster area is handled by the task force. The two change descriptions require changes of this process if standard processing is not possible. On the one hand, it might happen that the transport of the equipment is delayed as customs might not clear the equipment immediately. On the other hand, equipment transport capacity might not be available right away. For both cases, the process of transporting the equipment has to be changed accordingly.

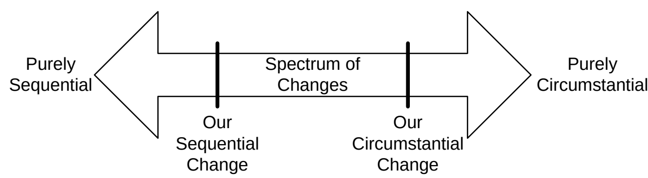


**Fig. 2.** Change Types

**Factor and Factor Levels**: The considered factor in our experiment is the type of the change task with factor levels *sequential* and *circumstantial*. It is important to note that the two change tasks used in the experiment are not strictly sequential and circumstantial. However, when compared to each other, one change is clearly more sequential, or circumstantial, respectively, than the other (cf., Fig. 2). We also ensured that both changes require the same effort in terms of graph-edit distance (i.e., the minimal number of atomic graph operations needed to transform one graph into another, it can be leveraged to assess the similarity of two process models [5]). For both changes, the graph edit distance between the original model and the changed model, i.e., the number of operations needed to perform the change is around 40 atomic change operations.
**Response Variables**: As response variables we consider the *modeling speed* of conducting the modification tasks, the *accuracy* of the change, the *correctness*

of the resulting model as well as the perceived *cognitive load* of conducting the modification tasks. *Modeling speed* is measured as time (in seconds) needed for conducting a change task. For assessing the *accuracy* we utilize a set of 12 key properties for each change type, which are derived directly from the corresponding change description. For instance, "in the meantime" indicates parallel execution, whereas explicit naming of activities in the text indicates that respective activities should also be present in the process model. One point is rewarded for each fulfilled property in the solution model (e.g., existence of parallel execution). In addition, accuracy also includes penalty points for negative key properties (e.g., superfluous activities). Consequently, students are able to gather at most 12 points per change, allowing us to quantify their models in terms of accuracy. *Correctness*, in turn, is assessed in terms of model syntax as well as execution semantics. That is, whether syntactic requirements imposed by the BPMN specification are met, and whether the model is free of behavioral anomalies such as a deadlock or a lack of synchronization. To this end, we applied the well-known soundness criterion [19]. For obvious reasons, soundness checking is done solely for syntactically correct models. Finally, subjects are asked to assess their *cognitive load* (i.e., the perceived difficulty of conducting a change task) on a 7-point Likert scale.

**Hypothesis Formulation:** Goal of the experiments is to investigate whether the type of change is influencing *modeling speed*, *accuracy*, *correctness*, and *cognitive load*. Accordingly we postulate the following hypotheses:

- **Null Hypothesis** $H_{0,1}$: There is no significant difference in the modeling speed of conducting a process changes depending on the type of change.
- **Null Hypothesis** $H_{0,2}$: There is no significant difference in the accuracy of the resulting models with respect to the type of change.
- **Null Hypothesis** $H_{0,3}$: There is no significant difference in the correctness of the resulting models with respect to the type of change.
- **Null Hypothesis** $H_{0,4}$: There is no significant difference in the perceived cognitive load depending on the type of change.

**Instrumentation**: The participants conducted the modeling using the Cheetah BPMN Modeler [4] which is a graphical process editor. The editor provides only basic drawing functionality for creating, moving, and deleting nodes and edges of a single BPMN diagram; the modeling constructs were limited to tasks, start and end events, gateways (AND, XOR), and control flow edges. The reduced functionality mimics a flexible "pen and paper" setting. To be able to trace the actual modeling process, we extended the BPMN Modeler with a logging function, which automatically records every modeling step and allows us to derive performance characteristics (e.g., modeling time, number of syntactical errors, number of events) for each model, and a function to replay a modeling log.

**Experimental Design**: The experimental setup is based on literature providing guidelines for designing experiments [22]. Following these guidelines a *randomized balanced single factor* experiment is conducted with *repeated measurements*. The experiment is called *randomized* because subjects are assigned to groups randomly.

We denote the experiment as *balanced* as each factor level is used by each subject, i.e., each student works on a sequential and circumstantial change task. As only a single factor is manipulated (i.e., the change type), the design is called *single factor*. Due to the balanced nature of the experiment, each subject generates data for both factor levels and thus provides *repeated measurements*. Figure 3 depicts the design following the aforementioned criteria. The subjects are randomly assigned to two groups of equal size, subsequently referred to as Group 1 and Group 2. To provide a balanced experiment with repeated measurements, the overall procedure is divided into two runs. In the first run Group 1 works on a *sequential* change task, Group 2 on a *circumstantial* one. In the second run factor levels are switched and Group 1 applies factor level *circumstantial*, Group 2 factor level *sequential*. Since no subject deals with an object more than once this design avoids learning effects.

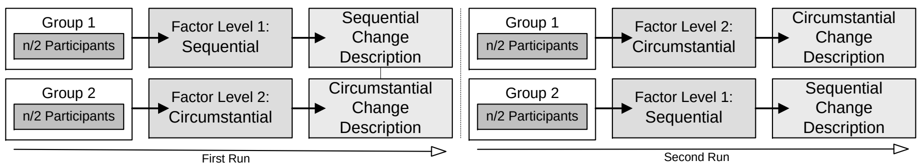


**Fig. 3.** Employed Experimental Design

## 4 Performing the Experiment

By now, the set-up of the experiment has been explained. Section 4.1 describes the preparation and execution of the experiment. Then, the analysis and interpretation of the gathered data are presented in Section 4.2. Finally, a discussion of the results is provided in Section 4.3.

### 4.1 Experimental Preparation and Execution

**Preparation**: As part of the experimental preparation, we created the model for the "Transport of Equipment" process and two different change task descriptions, one rather sequential change task and one rather circumstantial change task. In order to ensure that each description is understandable and can be modeled in the available amount of time, we conducted a pre-test with 14 graduate students at the University of Innsbruck. Based on their feedback, the change task descriptions were refined in several iterations; the resulting tasks are shown in Appendix A.
**Execution** The experiment was conducted in January 2010 in Potsdam. A session started with a familiarization phase, in which students had 10 minutes to investigate the given model for the "Transport of Equipment" process. At the end of the familiarization phase, students had to answer comprehension questions on the "Transport of Equipment" process before they were able to proceed with the experiment. The familiarization phase was followed by a modeling tool tutorial in which the basic functionalities of the BPMN Modeler were explained to our subjects. The students were then randomly divided into two groups. As pointed

out in Section 3, the experiment was executed in two subsequent runs. After completing the two change tasks, a questionnaire on cognitive load was presented to the students.

**Data Validation**: Once the exploratory study was carried out, the logged data was analyzed. Data provided by 15 students was used in our data analysis.

### 4.2 Data Analysis

In this section, we describe the analysis of gathered data and interpret the obtained results.

**Testing for Differences in Modeling Speed:** To test for differences in terms of modeling speed, a t-test for homogeneous variances was applied [13]. The test was applicable to analyze time differences because the samples of both factor levels follow normal distributions and the variances of the samples are homogeneous. With an obtained p-value of 0.818 ($> 0.05$), hypothesis $H_{0,1}$ cannot be rejected at a confidence level of 95%. In other words, there is no statistically significant difference with respect to the speed of answering between the two factor levels. This outcome is re-enforced by considering the boxplots in Fig. 4.

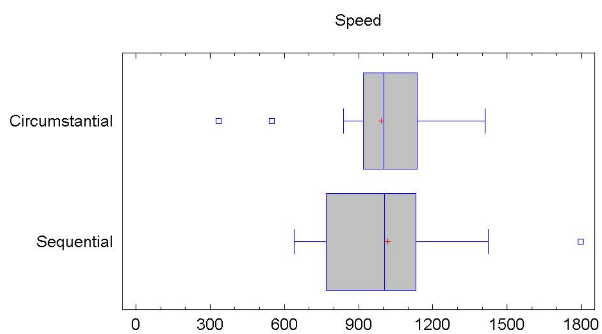


**Fig. 4.** Speed of Executing the Modeling Task

**Testing for Differences in Accuracy:** Fig. 5 shows the boxplots displaying the distribution of the accuracy values as obtained for the two factor levels, i.e., the circumstantial and the sequential change task. For the circumstantial task compared to the sequential task the median value is lower, as well as the overall distribution is being situated at the lower side of the accuracy axis. To test whether differences in terms of accuracy are statistically significant, we deploy the t-test. The test is applicable here because both samples are normally distributed and the variances of the samples are homogeneous. With an obtained p-value of 0.042 ($< 0.05$) hypothesis $H_{0,2}$ is rejected at a confidence level of 95%. In other words, the lower accuracy values obtained for the circumstantial task are statistically significant.

**Testing for Differences in Correctness:** To test for differences in correctness between the two factor levels, we inspected all models against the BPMN standard and scored whether these models were syntactically correct or not. Since the

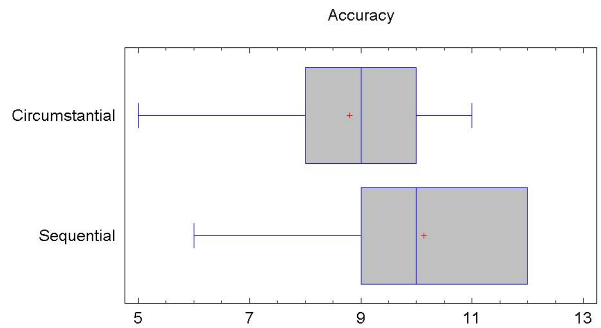


**Fig. 5.** Accuracy of Process Models

binomial data that was obtained in this way was not normally distributed, we applied the non-parametric Mann-Whitney test [18]. This resulted in a p-value of 0.053 ($> 0.05$). As an alternative way to compare the correctness of the models, we considered the *soundness* of the produced models, which is a well-established correctness notion for process models [19]. We applied the same statistical test to compare the two factor levels, which led to a p-value of 0.275 ($> 0.05$). Since both p-values exceed the threshold of 0.05, either narrowly or widely, the hypothesis $H_{0,3}$ cannot be rejected at a confidence level of 95%: No statistically significant differences with respect to correctness can be observed.

**Testing for Differences in Cognitive Load:** As stated before, we asked all respondents to rate the cognitive load of the two modeling tasks after they had been performed. We rated this complexity on a 7-point Likert scale, ranging from 'very low' to 'very high'. The values that were obtained in this way were in conformance with the requirements for a standard t-test. The application of this test resulted in a p-value of 0.735 ($> 0.05$). Consequently, hypothesis $H_{0,4}$ cannot be rejected at a confidence level of 95% or, phrased differently, no statically significant difference can be established between the cognitive load between the groups.

### 4.3 Discussion of Results

With respect to the four different performance indicators that were examined for differences, only accuracy indicates a significantly better performance for the group performing the sequential change task. In this case the obtained p-value of 0.042 is a bit below the cut-off value of 0.05. For all other indicators, i.e., correctness, speed, and cognitive load, no significant differences could be detected.

These outcomes point at the type of change not being an overly strong factor with respect to the maintainability of a process model. A significant difference is expected from a theoretical point of view, as the respondents were asked to carry out a change task on a process model that is captured with a technique that emphasizes a sequential view on the process. Therefore, we expected a change task that is captured in the same, sequential style to be performed easier or better than a circumstantial change task.

For the interpretation of these results we have to consider two major factors that we tried to neutralize. First, there are characteristics of the process modeling language that influence the ease of change. Arguably, BPMN process models can be rather easily changed in comparison to Petri nets, which require the alternation of places and transitions to be preserved. Accordingly, the size of our models in the experiment could have been too small for the effect of change type to materialize. Second, experiments like ours are strongly influenced by the process modeling expertise of the participants [15]. It might have been the case that our pre-test population was less proficient in process modeling, such that the selected models again could have been too simple for the experimental group.

There are alternative explanations. We chose on purpose change tasks of a different type while ensuring that the graph-edit distance for solutions to the sequential and circumstantial tasks are the same. This might also be a hint that the graph-edit distance could be a much stronger factor for approximating the difficulty of a change requirement. On the other hand, the number of respondents that has been involved in this experiment (15) is rather low, which makes statistical inferences hazardous due to a high impact of individual observations. Given such a small sample size we are only able to detect strong effects in the data. The impact of change type on accuracy seems to be such a strong effect. Finally, the familiarization phase during which all respondents could inspect the base model has been considerable. It could be argued that the remaining sense-making task (e.g., the interpretation of the change task) is a minor effort in the overall task. All these issues can only be settled satisfactorily by replicating this experiment with a larger respondent base, a shorter familiarization phase, and another set of change tasks.

## 5 Related Work

In this section we first discuss factors that influence the usability of process models and which we strived to keep constant. Then, we relate works to our experiment that investigate the impact of representational characteristics of a model on comprehension and maintainability.

There are several factors influencing process model usability including domain knowledge, tool support, and selection of tasks. Prior *domain knowledge* can be an advantage for participants of an experiment. People may find it easier to read a model about the domain they are familiar. It is known from software engineering that domain knowledge affects the understanding of particular code [11]. Its impact is neutralized in experiments by choosing a domain that is usually only known by experts. *Tool support* plays a fundamental role in fostering process changes and hiding the complexity behind high-level change operations [17, 21]. We tried to neutralize the impact of tool support by offering only the most atomic change operations. The *selection of experimental tasks* can also have an impact on the validity of an experiment. It has been shown that understanding tasks can vary in their degree of difficulty even if they relate to the same model [14]. We tried to neutralize the impact of the tasks by choosing tasks of equal graph-edit distance.

Our experiment can be related to various experiments that investigate how characteristics of a particular problem representation influences problem-solving performance. We have already referred to work on software program comprehension [7, 8, 9]. It showed that declarative programs are better at explicating circumstantial information while imperative programs more handily show sequential information. This work is particularly interesting as it contributed to settling a long debate on whether declarative or imperative computer programs should be considered to be superior. Confirming results are reported among others in [1, 10, 16] where the impact of a particular information representation is tested as a factor of comprehension performance. This exactly matches the more general argument of cognitive fit theory, which states that a problem representation should match the problem solving task [20].

## 6 Summary and Conclusion

In this paper we investigated the relationship between the type of a change requirement and the performance of modifying a process model. We designed and conducted an experiment in which graduate students received sequential and circumstantial change requirements and changed a BPMN model accordingly. The results show that there is partial support for the type of change being a factor for process model maintainability. Our findings are of significant importance to future experiments on business process maintainability. Apparently, the way a change requirement is formulated has an impact on the ease of changing the model. Experiments that do not investigate this effect must neutralize its impact either by using only one type of change requirement or by making a balanced selection of change tasks from both types.

In future research we aim to replicate this experiment with more students in a similar classroom setting. It will be interesting to check whether a larger sample size will reveal effects that have been too weak to be detected with our small sample. Furthermore, we plan to conduct experiments that vary the set of change operations that are offered to the modeler. While we currently provided only basic change operations in this experiment, it has to be investigated whether complex changes can be easily made once high-level change operations are available. This argument points also to the need for further research into change operations. We consider it an important question of how circumstantial change requirements can be directly translated into corresponding change macros.

## Acknowledgement

This research is supported by the Technology Foundation STW, applied science division of NWO, and the technology program of the Dutch Ministry of Economic Affairs.

## References

1. R. Agarwal, P. De, and A.P. Sinha. Comprehending object and process models: An empirical study. *IEEE TSE*, 25(4):541–556, 1999.
2. W. Bandara, G. Gable, and M. Rosemann. Factors and measures of business process modelling: model building through a multiple case study. *EJIS*, 14(4):347–360, 2005.
3. G. Canfora, F. García, M. Piattini, F. Ruiz, and C.A. Visaggio. A family of experiments to validate metrics for software process models. *JSS*, 77(2):113–129, 2005.
4. Cheetah. Website of the Cheetah Plattform. http:// cheetahplatform.org.
5. R. Dijkman, M. Dumas, and L. García Bañuelos. Graph Matching Algorithms for Business Process Model Similarity Search. In *Proc. BPM '09*, pages 48–63, 2009.
6. D. Fahland and H. Woith. Towards process models for disaster response. In *Proc. BPM '08*, pages 254–265, 2008.
7. D.J. Gilmore and T.R.G. Green. Comprehension and recall of miniature programs. *IJMMS*, 21(1):31–48, 1984.
8. T.R.G. Green. Conditional program statements and their comprehensibility to professional programmers. *Journal of Occupational Psychology*, 50:93–109, 1977.
9. T.R.G. Green and M. Petre. Usability analysis of visual programmingenvironments: A 'cognitive dimensions' framework. *J. of Vis. Lang. and Computing*, 7(2):131–174, 1996.
10. J. Hahn and J. Kim. Why are some diagrams easier to work with? effects of diagrammatic representation on the cognitive integration process of systems analysis and design. *ACMTCHI*, 6, 1999.
11. A. Lakhotia. Understanding someone elses code: Analysis of experiences. *JSS*, 23(3):269–275, 1993.
12. C. Li, M. Reichert, and A. Wombacher. On measuring process model similarity based on high-level change operations. In *Proc. ER'08*, pages 248–264, 2008.
13. L.A. Marascuilo and R.C. Serlin. *Statistical Methods for the Social and Behavioral Sciences*. Freeman New York, NY, 1988.
14. J. Melcher, J. Mendling, H. A. Reijers, and D. Seese. On measuring the understandability of process models. In *Proc. BPM '09*, 2009.
15. J. Mendling, H. A. Reijers, and J. Cardoso. What makes process models understandable? In *Proc. BPM '07*, pages 48–63, 2007.
16. 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, 1993.
17. M. Reichert and P. Dadam. ADEPT$_{flex}$ – Supporting Dynamic Changes of Workflows Without Losing Control. *JIIS*, 10(2):93–129, 1998.
18. S. Siegel and N.J. Castellan. Nonparametric Statistics for the Social Sciences. *McGraw-Hill*, 1988.
19. W.M.P van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 1998.
20. I. Vessey. Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*. *Decision Sciences*, 22(2):219–240, 1991.
21. B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - enhancing flexibility in process-aware information systems. *DKE*, 66(3):438–466, 2008.
22. C. Wohlin, R. Runeson, M. Halst, M.C. Ohlsson, B. Regnell, and A. Wesslen. *Experimentation in Software Engineering: an Introduction.* Kluwer, 2000.

# A Change Descriptions Used in the Experiment

*Sequential description.* Customs of the host country may deny clearance of equipment after presenting equipment at customs or after demonstration of devices. If equipment is not cleared by customs of the host country, the task force members try to convince customs officials to clear the equipment with incomplete documents. In the meantime, task force members contact their partners to trigger support from higher-ranked authorities of the host country. If the customs officials finally clear the whole equipment by negotiation and support, the equipment is transported to a storage location. In the other case, equipment is usually not cleared because of incomplete documents for some parts of the equipment. Those parts that have been cleared are transported to the storage location, whereas the missing documents for the remaining parts are retrieved from the office in Germany. Once these documents are available, the remaining parts of the equipment are transported to the storage location as well.

*Circumstantial description.* Usually, equipment transport capacity is not available immediately. Therefore, the process is adapted to ensure efficient handling of the equipment. The task force team members travel in split groups to the destination. A first group flies to the host country ahead of the equipment right away. After having presented itself at the immigration it takes care of road maps, renting of vehicles, and organizing accommodation. In the meantime, a second group handles all equipment logistics in Germany and then flies to the disaster area independently of the equipment. Eventually, the second group passes immigration and contacts the other task force team members. In the meantime, the second group also contacts local geologists, if there is a local institution with geologic know-how. The equipment is cleared in the host country as soon as it arrives. The whole equipment handling in the host country including customs is done by the second group of task force members. The first and the second team synchronize after their respective processes and transport the cleared equipment to the storage location.