Investigating the Usefulness of Notations in the Context of Requirements Engineering

Research Agenda and Lessons Learned

Anne Gross[1], Jakub Jurkiewicz[2], Joerg Doerr[1], Jerzy Nawrocki[2]
[1]Fraunhofer IESE
Kaiserslautern, Germany
{Anne.Gross, Joerg.Doerr}@iese.fraunhofer.de

[2]Institute of Computer Science
Poznan University of Technology, Poznan, Poland
{Jakub.Jurkiewicz, Jerzy.Nawrocki}@cs.put.poznan.pl

***Abstract*—In recent years, empirical studies have gained more and more importance in requirements engineering. Especially studies aimed at investigating the efficiency and effectiveness of software requirements specification techniques have been reported frequently. In fact, objective and quantifiable data collected during experimental investigations can be very beneficial both for researchers evaluating new methods and for practitioners, who have to decide which technique to choose within a certain context. However, in order to deliver sound and empirically valid data, experimental investigations have to be planned and conducted carefully. This is a challenging task, as it requires experimenters to think and decide about important aspects and control possible threats to validity. In this paper, the authors report about their experiences in jointly planning and conducting an experimental comparison of prominent notations such as UML Activity Diagrams (ACT), Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC), and Use Cases. These lessons learned are supplemented with parts of their current research agenda as well as the results they achieved by applying the experimental design in initial experiment runs. In future work, the aim is to plan and run further experimental comparisons by applying the design presented in this paper.**

***Keywords*-requirements specification; requirements specification techniques; experimental comparison; empirical studies; lessons learned; Event-driven Process Chains; Business Process Model and Notation; Use Cases; UML Activity Diagrams**

I. INTRODUCTION

In 2005, Sjøberg et al. [1] investigated over five thousand papers in the field of software engineering published during a period of ten years. They found only 103 descriptions of controlled experiments (1.9% out of the total number of investigated papers), with 18% of these 103 experiments being replications of previously conducted studies.

In recent years, experimental investigations as well as replications have gained more and more importance, also in requirements engineering (RE) [2]. In this particular field, empirical research has especially been targeted at investigating the efficiency and effectiveness of software requirements specification techniques, such as [3][4] [5].

Among these kinds of studies, comparisons of various requirements specification notations were conducted and reported with the intent to investigate their ability to (1) support certain RE-related activities such as business process analysis or requirements validation or (2) fulfill important aspects of software requirements specifications (SRS), such as completeness, understandability, or correctness [21] [22] [23] [24] [25] [26].

In fact, there exist a variety of notations that are at the disposal of practitioners today. Typically, it may be the user's familiarity with these that matters most when selecting a particular notation. However, objective and quantifiable data about the usefulness of certain notations delivered by experimental comparisons can be very helpful for practitioners who need to decide which notation to use in which context. Nevertheless, in order to benefit practitioners, such experimental investigations have to be planned, conducted, and analyzed carefully to ensure that the results are valid, can be interpreted clearly, and can be compared. This is a challenging task, which requires experimenters to think and decide about various important aspects and control possible threats to validity.

In order to achieve this task, *research questions* and *hypotheses* have to be defined that should be investigated in a particular study. This comprises, for instance, decisions about notations (i.e., *independent variables*) that should be investigated as well as properties that should be analyzed and compared between different notations (i.e., *dependent variables*). In the second step, suitable *metrics* have to be derived that allow measuring the defined properties. This activity is typically supplemented with the preparation of *study material* (e.g., guidelines, measurement instruments) that allow observing and collecting data related to the identified metrics and dependent variables during the investigation.

When planning an experimental investigation, another decision is related to the selection of *participants*. This decision actually has tremendous influence on the interpretation and generalization of the results to the community. Further activities comprise the definition of the *experiment procedure*, including the assignment of the participants to the treatments, as well as the definition of

suitable *strategies* for analyzing the collected data. Finally, the analyzed data has to be *interpreted* and discussed, taking into account possible threats to validity. At the end, all procedures and results must be *reported*.

In the past, the authors have been actively involved in planning and conducting empirical investigations of notations [7] [8] [9] [10] [11]. In 2010, the authors decided to jointly collaborate on an experiment aimed at investigating prominent business process modeling notations. The investigation was based on the study design of a previous investigation [10]. However, during this collaboration, the authors encountered several challenges; their lessons learned are summarized in this paper. Furthermore, the paper shares parts of their current research agenda as well as initial results they achieved by applying the experimental design in initial experiment runs. Especially the research agenda could be useful for other researchers and practitioners who also plan to conduct empirical studies of this kind. Using the same agenda would allow easier replication and comparison of experiments conducted by different researchers.

The remainder of this paper is structured as follows: Section II presents the activities, the research agenda, and the lessons learned related to the planning phase of an experimental investigation. Section III discusses the activities, the research agenda, and the lessons learned related to the actual running phase of the experiment. Section IV deals with strategies and lessons learned while analyzing and reporting the results. Section V is dedicated to presenting the initial results achieved in the first experiment runs as well as an outlook on future work. Finally, the paper concludes with a summary in Section VI.

II. PLANNING THE EXPERIMENT

Before an experiment can actually be conducted, effort has to be spent on planning the investigation thoroughly. This includes decisions about research questions, hypotheses, metrics, participants, etc. In the following, we report about our experiences related to the planning phase illustrated with extracts from our current research agenda.

A. Research Goals and Hypotheses

In order to state the research questions clearly and derive appropriate metrics, we consider the Goal/Question/Metric paradigm (GQM) [12] as being very helpful. Typically, the research question is stated in the overall research goal (G), which can be defined by using a GQM template [12] as illustrated in Table 1.

In our current research agenda, the research questions and goals are aimed at analyzing and comparing prominent business process modeling notations such as UML Activity Diagrams (ACT) [13] Event-driven Process Chains (EPC) [14], Business Process Model and Notation BPMN [15] [16], and Use Cases (UC) [17] with respect to their usefulness for typical requirements engineering tasks. As these tasks are manifold, we claim that the usefulness of these notations should be investigated from several viewpoints. Obviously, we consider the *viewpoint of a requirements engineer*, who has to use the notations to specify requirements in an SRS, as being very important.

However, the understandability of artifacts specified with any of these notations is also very crucial in requirements engineering. In fact, understandability has been reported as being the most commonly evaluated aspect of SRS [2][6]. Therefore, we also consider the *viewpoint of customers or end users*, who are typically responsible for validating the correctness of the requirements stated in the SRS as being worth investigating. Therefore, our current research agenda also includes metrics, hypotheses, material, etc. that will be used to investigate the viewpoint of the customers / end users in experiment runs. Furthermore, we also consider the *viewpoint of engineers* involved in downstream activities like architecture, design, coding, and testing as important for investigating the understandability of notations. Currently, we have not detailed our research agenda yet for this latter viewpoint, but this will be the subject of future work (see also Section V). Based on all these aspects, we can state our overall research goals as:

Table 1: Overall research goal (using GQM template)

Analyze *ACT, EPC, BPMN, and UC* for the purpose of *their comparison* with respect to *their usefulness* from the viewpoint of *requirements engineers and customers / end users.*

As the quality focus of this goal (i.e., the usefulness of the notations) is still on a high level, it has to be refined into sub-goals. Following the GQM paradigm we supplement this refinement by deriving questions and corresponding metrics (see Table 2 and Table 3) based on the experiment design published in [10]). Table 5 and Table 6 in Section II.B provide further details on the metrics.

Table 2: GQM Viewpoint of Requirements Engineers

Viewpoint of Requirements Engineers
G_1: Analyze *ACT, EPC, BPMN, and UC* with respect to *their efficiency,* which is reflected in the *complexity of the artifacts created with any of the investigated notations.* **Q_1**: Which of the notations produces less complex artifacts? **M_1**: number of elements required to specify information (process aspects) contained in a given process description **(# elements / # specified process aspects)**
G_2: Analyze *ACT, EPC, BPMN, and UC* with respect to *their efficiency*, which is reflected in the *time required to create artifacts with any of the investigated notations*. **Q_2**: Which of the notations allows creating artifacts faster? **M_2**: time required to specify information (process aspects) contained in a given process description **(time / # specified process aspects)**
G_3: Analyze *ACT, EPC, BPMN, and UC* with respect to *their effectiveness*, which is reflected in the *correctness of the created artifacts*. **Q_3**: Which of the notations produces more erroneous artifacts? **M_{3a}**: # of syntax errors (**# $errors_{svn}$**) **M_{3b}**: # of incorrectly modeled process aspects (**# $errors_{inc}$**) **M_{3c}**: # of missing process aspects (**# $errors_{miss}$**)

Table 3: GQM Viewpoint of customers / end users

Viewpoint of customer / end user
G_4: Analyze *ACT, EPC, BPMN, and UC* with respect to their *understandability*, which is reflected in the *understandability of the artifacts created with any of these notation*s. **Q_4**: Which of the artifacts is easier to understand? **M_4**: number of incorrect answers given by the participants to content-related questions referring to a given artifact (**# IAQ**).
G_5: Analyze *ACT, EPC, BPMN, and UC* with respect to their *understandability*, which is reflected in the *reliability of finding errors in given artifacts created with any of these notations*. **Q_5**: Which of the notations supports more reliable identification of errors? **M_5**: number of incorrect answers given by the participants while finding errors in erroneously specified artifacts (**# IAE**)

The definition of goals is also supplemented with the definition of hypotheses that should be investigated. Typically, hypotheses are specified as alternative hypotheses (H_1) and null hypotheses (H_0). While the alternative hypothesis formalizes our research hypothesis (i.e., what we want to show), the null hypothesis corresponds to a general or default position. This position basically reflects the opposite situation of our research hypothesis. Testing the null hypothesis is required in order to back up our results by showing that no relationship that appears in our data was produced by random chance. In fact, there is a strong interrelation between the questions, metrics and hypotheses as both the questions and related hypotheses reflect dependent variables we want to observe in our study (e.g., understandability of artifacts, complexity of artifacts). These variables are again operationalized by the metrics.

Table 4 illustrates hypotheses that are the subject of our current research agenda on two examples. All other hypotheses related to the goals stated above can be formalized accordingly using the same scheme.

Table 4: Exemplary Hypotheses

Hypotheses (Examples) related to G_3 and G_5
$H_{3\text{-}0}$: There is *no difference* between the notations regarding errors between artifacts created with any of the investigated notations with errors $\in$ {$errors_{syn}$, $errors_{inc}$, $errors_{miss}$}. **$H_{3\text{-}1}$**: There is *a difference* between the notations regarding errors between artifacts created with any of the investigated notations with errors $\in$ {$errors_{syn}$, $errors_{inc}$, $errors_{miss}$}.
$H_{5\text{-}0}$: There is *no difference* between the notations regarding the number of errors found by the participants in erroneously specified artifacts (IAE). **$H_{5\text{-}1}$**: There is *a difference* between the notations regarding the number of errors found by the participants in erroneously specified artifacts (IAE).

B. *Metrics and Study Material*

This section provides detailed information about the metrics introduced previously as well as prepared study material.

Table 5: Metrics (Requirements Engineer)

Viewpoint of Requirements Engineer
M_{1a}: # of specified process aspects: For each participant it is examined whether all aspects included in a given process description (see Section II.B.1) have been specified in the artifacts produced by the participants during the experiment. This metric contains the total number of specified process aspects, regardless of whether they are specified correctly or incorrectly.
M_{1b}: # of elements: This metric enables to draw conclusions with respect to the complexity of the artifacts created with any of the investigated notations. To measure this metric, the number of elements in the created artifacts can be counted. Examples of elements include symbols like events, functions, roles, pools, logical operators, data objects, message flows, etc.
M_2: time: This metric contains the time required by the participants to create artifacts with any of the investigated notations.
M_{3a}: # of $errors_{syn}$: This metric contains the number of notation-specific syntax rules that are neglected related to a number of available rules of interest in an investigation. Examples of such rules include "An ACT always has to start with a start event", "In an EPC each function and event may only have one incoming and one outgoing connector", "In BPMN message flows cannot connect elements that exist in different pools", "All steps in the UC scenarios clearly indicate whether the system or actor performs the step".
M_{3b}: # of $errors_{inc}$: This metric contains the number of process aspects included in the given process description, that were addressed by the participants in the artifacts they created, but which are semantically incorrect. Examples include: incorrect preconditions, incorrect events, incorrect roles assigned to activities.
M_{3c}: # of $errors_{miss}$:: This metric contains the number of process aspects included in the given process description that are not addressed at all in the artifacts created with any of the investigated notations.

Table 6: Metrics (Customer / End User)

Viewpoint of Customer / End User
M_4: # IAQ: This metric contains the number of incorrect answers given by the participants while answering specific questions related to the content of a given artifact (examples can be found in Section II.B.2)
M_5: # IAE: This metric contains the number of incorrect answers given by the participants while identifying errors in an erroneously specified artifact. This metric comprises errors that are not identified at all as well as wrongly identified errors (examples can be found in Section II.B.2).

In the following, we present some information about our study material comprising (1) study artifacts the participants will basically work with during the experiment, (2) guidelines that capture information related to the experiment procedure, and (3) measurement instruments that have been prepared to collect relevant data.

1) Study Artifacts: In order to investigate the **viewpoint of the requirements engineer** in experiment runs, we prepared a scenario description, i.e., a textual description of a given process that has to be specified by the participants using one of the investigated notations. This process describes a daily routine of a caregiving organization providing care services for elderly persons supported by a so-called "Digital Care Giver Assistant" (DCGA) – a mobile device that provides the caregiver with patient-related information received from sensors mounted in the patient's apartment. The process comprises a total of 15 process aspects that should be specified by the participants. An example of such an aspect is: "As soon as she (the caregiver) arrives at Ms. Schmidt's apartment, the DCGA automatically shows a list of all care tasks she has to perform".

The main study artifacts to be used in experiment runs investigating the **viewpoint of the customer / end user** are *two process specifications*, which were created with the each of the investigated notations. One of the processes specifies a process of preparing and celebrating a garden party (used as input for task 1, see guidelines below). The second process description represents an erroneously specified process description related to the regulations of a European Soccer Championship (used as input for task 2, see guidelines below). It needs to be noticed that the introduction of errors may influence the results of the experiment; therefore, we introduced only a small number of changes that were only minor and not obvious at first glance (e.g., wrong post-conditions in events); furthermore, trial experiments could help to assure that the introduced errors can give valid results. This second process description is supplemented with a *correct textual description* of a process related to the regulations of a European Soccer Championship (also used as input for task 2, see guidelines below).

2) Guidelines: For the investigation of the **requirements engineer's viewpoint** we prepared a problem description as a guideline, which provides detailed background information related to the work context of the caregiver, the domain, the involved systems, etc. Additionally, a guideline for the participants was created, which instructs the participants how to run through the experiment (e.g., start reading the problem description, then specify given process aspects, fill out the questionnaire (see also Section II.B.3) at the end).

For experiment runs investigating the **customers' / end users' viewpoint**, we prepared *two task descriptions* as guidelines. The *task description (Task #1)* has been designed to investigate the understandability of the notations with the help of metric #IAQ (see Table 6). In this first task description, the participants are asked to answer 14 content-related questions regarding the given process of preparing and celebrating a garden party. Example: "When does the party end?", "Is it possible that guests are invited several times? If yes, why?" The second *task description (Task #2)* has been prepared to investigate the understandability of the notations with the help of metric #IAE (see Table 6). In this task description the participants are asked to identify errors in an erroneously modeled process description when comparing it to the given and correct process description in textual form (see the study artifacts introduced previously). Examples of errors in the diagram comprise incorrect usage of decision operators or events to reflect incorrect rules related to the qualification of teams.

Finally, a *guideline for the experimenters* was prepared in order to ensure that the conditions were kept the same during the experiments, as they are conducted at different locations (see also Section III).

3) Measurement Instruments: Finally, for each of the investigated viewpoints, a questionnaire was prepared as a measurement instrument for controlling possible influence factors such as participants' experience with the investigated notations, application domain, understandability of the scenario, etc. The questions comprised both open and closed questions. An example of a closed question: "Please indicate your experience with Event-driven Process Chains!" Alternative answers: "I never heard about it", "I heard about it but only in theory", "I heard about it and worked with it only during class", "I worked with it also outside class", "I work with it daily". In addition to controlling influence factors, the questionnaire was also designed to capture background information about the participants, such as age, gender, or education background (examples: "Please indicate you education background", "Please specify your topic of specialization (e.g., software engineering, requirements engineering)". Additionally, we created a spreadsheet that supported the analysis of metrics related to errors $\in \{errors_{syn}, errors_{inc}, errors_{miss}\}$ which we will introduce in Section IV.B as part of the lessons learned related to followed analysis strategies.

C. Participant Selection

One crucial activity during the planning phase is the selection of participants. In fact, the participants have to be selected carefully in order to draw valid conclusions about the outcome of the experimental investigation.

The participants must thus be selected in such a way that they appropriately represent the characteristics of the different viewpoints that are the subject of a particular empirical study and object of the research questions and hypotheses respectively. In our case we decided to recruit computer science students trained in requirements engineering, in particular in specifying requirements using the investigated notations, as subjects for investigating the

viewpoint of requirements engineers. For the investigation of the customers' / end users' viewpoint, we decided to recruit subjects unfamiliar with any of these notations, as this represents a typical situation in requirements engineering.

D. Lessons Learned Related to Experiment Planning

In the following, we will summarize our experiences and lessons learned (LL) related to the planning phase of empirical investigations aimed at investigating the usefulness of notations in the RE context.

LL_{Plan1}: If you select tools for specifying processes (as in the case of the requirements engineer's viewpoint), *make sure that the tools are as similar as possible and that these tools do not require any knowledge by the participants to be used properly*. In our initial experiment runs (see also Section V), we decided, for instance, to use a special tool (called "Bizagi") for modeling BPMN process models. The BPMN group was already familiar with this tool, as they also used it during their classes. Use Cases were specified with a text processing tool and EPC were created by hand on a sheet of paper. Even though all participants were familiar with the tools, the interpretation and analysis of errors was negatively influenced, especially when analyzing and interpreting $errors_{syn}$ (as some tools support rule checking) and the time required for creating artifacts. In the future, we will not use any modeling tools. Instead, we will simply use pen and paper to keep the conditions as similar as possible.

LL_{Plan2}: If you provide a scenario that is to be specified by the participants, *make sure that the given scenario is understandable by the participants*. Also be careful if the scenario is provided in a language that is not the native language of the participants. This may influence the understandability of the scenario. In any case, control the understandability of the scenario in the questionnaire by asking a dedicated question, such as "Was the provided scenario easy to understand?"

LL_{Plan3}: If the experiment is conducted at different locations and if you create objects like process descriptions (such as the DCGA scenario introduced in Section II.B.1), *consider involving all experimenters in creating the scenario description*. You could even consider creating several scenarios (one at each location) and assigning them randomly to the participants. As we based our research agenda on a previous design published in [10], the DCGA scenario had already been created at one location, i.e., in Kaiserslautern, Germany. In fact, after our replication during the initial experiment runs, the reviewers considered it as a threat to validity that the scenario was defined at only one location. This fact could bias the results, as the scenario would probably be more understandable or known to the German participants than to the participants from Poland. In future experiment runs, we will consider creating different scenarios (one at each location) or reuse scenarios from other similar studies.

LL_{Plan4}: *Assure equal power between different notations*. This is very crucial, for instance, if a given process description is specified (as in the case of the requirements engineer's viewpoint). That is, it has to be assured that each notation offers all elements required to model all aspects of the processes that have been considered in the experiment.

LL_{Plan5}: *Calculate sufficient sample sizes* already in the early planning phase by performing a power analysis [18]. Such an analysis can be used to calculate the minimum sample size required so that it is reasonably likely to detect an effect of a given size. Power analysis can also be used to calculate the minimum effect size that is likely to be detected in a study using a given sample size, a given alpha and a given power as well. It should also be used when testing the hypotheses.

LL_{Plan6}: When preparing objects such as process descriptions, it is important to *assure that the participants are familiar with the respective application domain* (e.g., embedded systems domain, information systems domain) *and do not require any special background knowledge* do understand the process. In order to control this influencing factor, the questionnaire should be used to capture possible problems related to the understandability of the domain and the provided material.

LL_{Plan7}: Especially for the requirements engineers' viewpoint, *it is very important for the participants to have experience with the notation they are assigned to*. Moreover, it has to be assured that *the experience with the investigated notation is equal among the participants*. This should be addressed by (1) providing sufficient training (for example in lectures) and (2) controlling the experience in the questionnaire (see also Section II.B.3). If neither the participants nor their experience are well known in advance, we recommend running some initial tests to establish the initial level of knowledge, e.g., to balance the group. We also recommend preparing a short hand out for each of the investigated notations, summarizing syntax, semantics, and rules to consider.

LL_{Plan8}: In our initial experiment runs, we faced the challenge of assigning the treatments (i.e., the notations) to the participants. In these initial runs, we particularly investigated three notations (BPMN, EPC, and UC) and assigned the notations to students at three different universities where the students were taught one of the three notations. That is, the BPMN notation was assigned to a group of students enrolled in a BPMN class in Atlanta (Georgia, USA); the EPC notation was assigned to a group of students who were taught EPC as part of a requirements engineering course at the University of Kaiserslautern (Germany); and the UC notation was assigned to a group of students at Poznan University of Technology in Poland who were taught the use of the UC notation to model business processes. However, this assignment was considered as a major threat to validity by different reviewers as the assignment was not done randomly, which might influence the generalizability of the results (because we were unable to assure equal training due to the different lectures). In the future, we intend to address this threat by *randomly assigning the notations to all participants independent of the university* – and providing additional training for the other notations.

LL_{Plan9}: In general, the selection of students as participants is often considered as a threat to validity, since

they are not comparable to real experts. On the other hand, there is always the argument to have as many participants as possible. But in fact, conducting an experiment with about 50 experts and at the same time controlling that they have the same experience is not an easy task. According to [19], selecting students is not different from selecting professionals if the complexity of the task does not overwhelm the students and if they receive the necessary training. Therefore, *we consider the recruitment of students not critical if sufficient training is provided and if the provided objects are on an appropriate level of difficulty*. We strongly recommend to additionally control the appropriate training, for example in the questionnaire.

III. Running the Experiment

A. Procedures and Treatment Assignment

This section discusses how an experiment should be run in order to avoid possible threats and to receive high-quality results. Table 7 introduces an experiment procedure that is part of our current research agenda.

Table 7: Experiment Procedure

Viewpoint of Customer / End User	
1.	Introduce the participants to the experiment. Present the goals of the experiment, the procedure, and the expected outcome of the participants' tasks.
2.	Assign participants to the treatments. If possible, the assignment should be done randomly, e.g., by drawing a number from a box.
3.	Assign a unique ID to every participant. This should make it easy to manage the results and compare them with the answers from the questionnaires. In case that several artifacts or documents are used during the experiment, ask the participants to write down the assigned ID on all the artifacts.
4.	Present the experimental material to the participants.
5.	Start the experiment by asking participants to follow the procedure according to the corresponding guideline. Ask the participants to write down the time when they start performing the task.
6.	Allow the participants to ask questions during the experiment. If they do, record all the questions and all the answers given.
7.	When a participant finishes his/her task or when the given time is up, ask the participants to write down the current time and hand in the created artifacts.
8.	At the end of the experiment, ask the participants to fill in the questionnaires.

B. Lessons Learned Related to Running the Experiment

Based on our experience, we can share the following lessons learned related to running the experiment:

LL_{Run1}: Make sure that *the time for performing the task is sufficient*. Restricting the time might influence the results as participants may start to hurry and compromise the quality in order to finish the given task. We recommend running trial experiments in order to see how much time is necessary to finish the task. In addition, the questionnaire should be used to control whether the time was sufficient.

LL_{Run2}: Create a *guideline that defines the procedure and give it to the participants*. This should prevent questions from the participants and ensure that the participants work in a similar manner.

LL_{Run3}: Make sure that *the conditions and the procedures in the groups are as similar as possible*. Following the experiment guideline (see Section II.B.2), assure the same conditions (both related to the experiment procedure and environmental factors) for all the participants even if the experiment is conducted at different locations. Document all events that interrupted the procedure, as they may influence the interpretation of the results. Consider following the procedure from previously conducted experiments as this might give you a chance to compare your results with that experiment.

LL_{Run4}: Make sure that *all the created artifacts and questionnaires are collected in the end*. Furthermore, if the participants were instructed to document information on the created artifacts (such as start and end times or ID) *check whether all artifacts and material contain this information*. If any of this information is missing, you might be forced to reject some of the artifacts from the analysis.

LL_{Run5}: Be careful *not to influence how the participants solve their task by either intentionally or even unintentionally providing them with information or hints regarding the solution*. To support this, either double-blind experiments could be conducted or researchers be asked to serve as experimenters that don't have any interest in the research goals and hence the results of the investigation.

IV. Analyze and Report Results

A. Analysis Strategies

This section is dedicated to the activities and lessons learned related to analyzing the collected data and reporting the results. During the analysis phase, each hypothesis that was defined during the planning phase (see Section II.A) will be investigated. This is achieved by first consolidating and preparing all collected and observed data during the experiment. Once all data has been consolidated, statistical analysis can be applied to determine any effects between the participants on the data regarding the hypotheses. That is, data can be analyzed by means of descriptive statistics, such as min-values, max-values, means, medians, etc.

In order to investigate whether there is a significant difference between the investigated notations, we test in our investigation equality of the means by performing a 2-tailed t-test. The t-test will also be supplemented by a Levene's test, which assesses the equality of variances in a given data set [27]. The significance level is set to $\alpha = 0.05$ for all tests.

Once all hypotheses have been investigated, a report has to be created that documents all results of the experimental investigation, including an interpretation and discussion of the results.

B. *Lessons Learned Related to Analyzing and Reporting the Results*

$LL_{Analyze1}$: In order to support the analysis and consolidation of data related to errors ∈ {$errors_{syn}$, $errors_{inc}$, $errors_{miss}$}, we recommend to use an analysis sheet as illustrated in Figure 1. This sheet was very helpful for us during initial experiment runs. In the column on the left, we documented all 15 process aspects that must be specified (see also Table 5). Based on that sheet we can analyze for each participant whether a given process aspect has been modeled correctly, has been modeled incorrectly ($errors_{inc}$), or is even missing ($errors_{miss}$). By using a color scheme, we mark the corresponding cells either in green (correct), orange (incorrect), or red (missing). Based on this color scheme we are finally able to calculate the metrics for each participant:

- # of specified process aspects (# of green cells + # of orange cells)
- # $errors_{inc}$ (# of orange cells)
- # $errors_{miss}$ (# of red cells)

Process Information				
Did the partipant specify that...	1	2	3	4
... Annette's DCGA receives a list of care takers she has to visit from the carecenter?	yes	other (see comment)	other (see comment)	other (see comment)
... the list of care takers is managed by an operator at the care center?	no, req is missing	other (see comment)	no, req is missing	no, req is missing
... as Annette arrives at the care takers apartment, the DCGA automatically shows a list of all care tasks she has to perform?	other (see comment)	other (see comment)	yes	yes

Figure 1: Analysis Sheet ($errors_{inc}$ and $errors_{miss}$)

$LL_{Analyze2}$: For $error_{syn}$, we follow a similar strategy by using a sheet capturing all relevant rules that have to be considered during modeling (see Figure 2). For each rule we then investigate whether the rule has been considered by the participant (green) or not (red).

29	Did the participant ...	22	4
30	... only draw sequence flows among events, processes, or gateways within the same pool?	yes	yes
31	... only draw message flows between events, processes, or gateways that exist in different pools (or between pools if 'black box')?	yes	yes
32	... make sure that start events do not have incoming sequence flow or outgoing message flow?	no (see comment)	yes

Figure 2: Analysis Sheet ($errors_{syn}$)

$LL_{Analyze3}$: When consolidating observed and collected data like $errors_{inc}$, $errors_{syn}$, $errors_{miss}$, we recommend that the experimenters perform this data collection independently first and discuss and consolidate the results afterwards. This procedure increases the reliability of the results.

$LL_{Report1}$: When reporting the results, provide descriptions of statistical terms (in case of space limitations, you might also consider including the definitions in a report comprising the material).

$LL_{Report2}$: Make the material and the raw data available online. This enables other researchers to replicate the experiment.

$LL_{Report3}$: When reporting the results, provide examples of metrics and dependent variables, e.g., missing process aspects.

$LL_{Report4}$: When reporting the results, discuss your results compared to existing approaches and your own research questions.

$LL_{Report5}$: When reporting the results, provide background information about the investigated notation (in case of space limitations, you might also consider including the definitions in a report comprising the material).

$LL_{Report6}$: Provide information related to the experience level, background, gender, and age of the students, as this information is important for other researchers or practitioners to judge whether the sample population fits the target group.

$LL_{Report7}$: To report the results, we recommend following the guidelines in [20].

V. Initial Results and Next Steps

As mentioned in the introduction, our current research agenda has already been applied in initial experiment runs. In these runs we investigated the goals related to the viewpoint of the requirements engineer (in particular G_1 to G_3 introduced in Table 2). Initially, we only compared three notations, i.e., BPMN, EPC, and UC, as we had access to three groups of students who were taught these notations. The analysis revealed significant differences between these notations in the context of business modeling from the viewpoint of a requirements engineer. For instance,

- the EPC group required significantly more elements to model certain process aspects compared to the BPMN group (complexity);
- the BPMN group produced significantly more $errors_{miss}$ compared to both the EPC and the UC groups.

Unfortunately, the weaknesses pointed out in the lessons learned sections (such as the assignment to the treatments) prevent us from drawing strong conclusions about the performed comparisons at this point in time.

As also noted previously, our research agenda is based on an earlier and already published experiment [10], which investigated all goals related to the viewpoints of requirements engineers as well customers. In this experiment, EPC and ACT were compared. We achieved significant results for the hypothesis that ACT are more effective and efficient than EPC from a requirements engineer's viewpoint. For the customer's / end user's viewpoint, we were not able to make a clear statement as to which of the two notations is more effective.

In the future, we plan further experiment runs comparing BPMN, EPC, UC, and ACT based on the research agenda introduced in this paper, taking into account our lessons learned. Furthermore, we plan to extend our research agenda to investigate the understandability of notations from the viewpoint of downstream development engineers (such as software architects, usability experts, testers) that base their development tasks on artifacts specified with these notations. Furthermore, we intend to define metrics and instruments to investigate the ability of the notations in terms of maintainability and changeability, which are also crucial aspects related to the efficiency of notations in software development projects from a requirements engineer's viewpoint.

VI. CONCLUSION

Empirical investigations in the context of RE have become more and more popular. In this particular field, empirical research has especially been targeted at investigating the efficiency and effectiveness of software requirements specification techniques. In fact, due to the increasing number of available notations, empirical evidence by means of objective data can be very helpful for practitioners who have to decide which notation to use in which context. However, in order to provide reliable data, such empirical studies have to be planned, conducted, and analyzed carefully, which is a very challenging task. In this paper, some lessons learned are shared that the authors have experienced when jointly planning and conducting an experimental comparison of prominent business process modeling notations. Furthermore, the paper introduces a research agenda that will be followed in future experiment runs. This research agenda as well as the lessons learned may also be very helpful for other researchers and practitioners who plan to conduct empirical comparisons of this kind or plan to replicate experiments.

ACKNOWLEDGMENT

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the EMERGENT project under grant no. 01IC10S01A, and partly funded by the Polish National Science Center project UMO-2011/01/N/ST6/06794 conducted at Poznan University of Technology.

REFERENCES

[1] D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, and N. K. Liborg "A survey of controlled experiments in software engineering". IEEE Transactions on Software Engineering 31(9):733–753, 2005

[2] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor: "Research Findings on Empirical Evaluation of Requirements Specification Approaches" In: 12th Workshop on Requirements Engineering, 16-17 July 2009, Valparaiso, Chile

[3] A. M. Davis, Ó. Dieste, A. M. Hickey, N Juristo, and A. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review". 14th International Conference on Requirements Engineering, IEEE Computer Society, 11-15 September 2006, Minneapolis, USA, pp. 176-185.

[4] Ó. Dieste, M. López, and F. Ramos: "Obtaining Well- Founded Practices about Elicitation Techniques by Means of an Update of a Previous Systematic Review". Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'2008), San Francisco, USA, July 1-3, 2008, pp. 769-772

[5] K. Ahmed, "A Systematic Review of Software Requirements Prioritization", Master Thesis in Software Engineering, School of Engineering Blekinge Institute of Technology, Sweden, 2006.

[6] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor: "A systematic Mapping Study on Empirical Evaluation of Software Requirements Specification Techniques" In: Third International Symposium on Empirical Software Engineering and Measurement, 15-16 Oct 2009, Buena Vista, US.

[7] B. Alchimowicz, J. Jurkiewicz, M. Ochodek, and J. Nawrocki, "Building Benchmarks for Use Cases", Comp. and Informatics, 2010

[8] M. Ochodek, B. Alchimowicz, J. Jurkiewicz, and J. Nawrocki, "Improving the reliability of transaction identification in use cases", Information and Software Technology Vol. 53, 2011.

[9] A. Ciemniewska, J. Jurkiewicz, L. Olek and J. Nawrocki, "Supporting Use-Case Reviews" . In Proceedings of 10th International Conference on Business Information Systems, 2007

[10] A. Gross and J. Doerr, "EPC vs. UML Activity Diagram - Two Experiments Examining their Usefulness for Requirements Engineering", In proceedings of Requirements Engineering Conference, p.47-56 , 2009

[11] I. Menzel, M. Mueller, A. Gross, and J. Doerr: "An Experimental Comparison Regarding the Completeness of Functional Requirements Specifications". In Proceedings of the 2010 18th IEEE International Requirements Engineering Conference (RE '10). IEEE Computer Society, Washington, DC, USA, 15-24., 2010

[12] V.R. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Paradigm", in: Encyclopedia of Software Engineering, Volume 1, pp. 528-532, John Wiley & Sons, 1994.

[13] D. Marlon and A. ter Hofstede, "UML Activity Diagrams as a Workflow Specification Language" in UML» 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools", pp. 76-90, 2001

[14] G. Keller, M. Nüttgens, and A.W. Scheer, "Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)", Universität des Saarlandes , 1992

[15] BPMN Business Process Model and Notation (BPMN), Version 2.0, January 2011, http://www.omg.org/spec/BPMN/2.0/PDF/

[16] S.A. White and D. Miers, "BPMN Modeling and Reference Guide - Understanding and Using BPMN", Future Strategies Inc., Lighthouse Pt, FL, ISBN-13: 978-0977752720, 2008

[17] Jacobson, I.: Use cases -- Yesterday, today, and tomorrow, in Software and systems modeling, pp 210-220, 2004

[18] J. Cohen, "Statistical power analysis for the behavioral sciences", (rev. ed.), Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc. , 1977

[19] W.F. Tichy, "Hints for Reviewing Empirical Work in Software Engineering," Empirical Software Engineering, vol. 5, pp. 309-312, 2000.

[20] A. Jedlitschka, M. Ciolkowski, and D. Pfahl "Reporting experiments in software engineering"; In: Shull F, Singer J, Sjøberg D (eds) Guide to advanced empirical software engineering, Chapter 8. Springer, London, 2008

[21] T. Koenig, T. Olsson, K. Schmid, and S. Adam, "Influence of Requirements Specification Notation on Change Impact Analysis. An Empirical Investigation", Research Report, Fraunhofer IESE, Kaiserslautern , 2007

[22] K. Kruczynski, "Business process modelling in the context of SOA – an empirical study of the acceptance between EPC and BPMN", World Review of Science, Technology and Sustainable Development, Vol. 7, No.1/2 pp. 161 – 168, 2010

[23] E. Rolón, J. Cardoso, F. García, F. Ruiz and M. Piattini, "Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models", Lecture Notes in Business Information Processing, Volume 29, 2009

[24] D.Q. Birkmeier, S. Klöckner, and S. Overhage, "An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users" ECIS 2010 Proceedings. Paper 51, 2010

[25] B. Bernárdez, A. Durán, and M. Genero, "Empirical Evaluation and Review of a Metrics - Based Approach for Use Case Verification", Journal of Research and Practice in Information Technology, Vol. 36, Number 4, 2004

[26] K. T. Phalp, J. Vincent, and K. Cox, "Improving the quality of use case descriptions: empirical assessment of writing guidelines", Software Quality Journal , Volume 15 Issue 4, 2007

[27] R. J. Carroll, H. Schneider, "A note on levene's tests for equality of variances", Statistics & Probability Letters, Volume 3, Issue 4, Pages 191-194, 1985