Exploring usability-driven Differences of graphical Modeling Languages: An empirical Research Report

Christian Schalles, John Creagh, Michael Rebstock*

Department of Computing
Cork Institute of Technology
Rossa Ave
Cork, Ireland
christian.schalles@mycit.ie, john.creagh@cit.ie

*Faculty of Economics and Business Administration
Hochschule Darmstadt University of applied Sciences
Haardtring 100
64295 Darmstadt, Germany
michael.rebstock@h-da.de

Abstract: Documenting, specifying and analyzing complex domains such as information systems or business processes have become unimaginable without the support of graphical models. Generally, models are developed using graph-oriented languages such as Event Driven Process Chains (EPCs) or diagrams of the Unified Modeling Language (UML). For industrial use, modeling languages aim to describe either information systems or business processes. Heterogeneous modeling languages allow different grades of usability to their users. In our paper we focus on an evaluation of four heterogeneous modeling languages and their different impact on user performance and user satisfaction. We deduce implications for both educational and industrial use using the Framework for Usability Evaluation of Modeling Languages (FUEML).

1 Introduction

In industry, models specifying information system requirements or representing business process documentations are developed by the application of various graphical modeling languages such as the UML and EPCs. In general, graphical modeling languages aim to support the expression of relevant aspects of real world domains such as information system structures or business processes [Lud03]. Almost all notations for software and business process specifications use diagrams as the primary basis for documenting and communicating them. The large number of available languages confronts companies with the problem of selecting the language most suitable to their needs. Beside functional and technical evaluation criteria, user-oriented characteristics of modeling languages are becoming more and more a focal point of interest in research and industry [SW07]. In this research paper we report about a comparative study on usability of selected modeling languages using the FUEML framework. The remainder of this paper is structured as follows: First, we analyze the theoretical background and state the hypotheses of this study. Secondly, we define usability in the domain of graphical modeling languages and additionally define metrics for measuring usability. Subsequently, we present our research methodology and our resulting findings. Lastly, we deduce implications based on our results and give an outlook on future research.

2 Theoretical Background

The variety of definitions and measurement models of usability complicates the extraction of capable attributes for assessing the usability of modeling languages. A usability study would be of limited value if it would not be based on a standard definition and operationalization of usability [CK06]. The International Organization for Standardization (ISO) defines usability as the capacity of the software product to be understood, learned and attractive to the user, when it is used under specified conditions [ISO06]. Additionally, the ISO defined another standard which describes usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [ISO98]. The Institute of Electrical and Electronics Engineers (IEEE) established a standard, which describes usability as the ease a user can learn how to operate, prepare inputs for, understand and interpret the outputs of a system or component [IEE90]. Dumas and Redish (1999) define, usability means quickness and simplicity regarding a users task accomplishment. This definition is based on four assumptions [DR99]: 1) Usability means focusing on users, 2) Usability includes productivity, 3) Usability means ease of use and 4) Usability means efficient task accomplishment.
Shackel (1991) associates five attributes for defining usability: speed, time to learn, retention, errors and the user specific attitude [Sha91]. Preece et al. (1994) combined effectiveness and efficiency to throughput [PRS$^+$94]. Constantine and Lockwood (1999) and Nielsen (2006) collected the attributes defining usability and developed an overall definition of usability attributes consisting of learnability, memorability, effectiveness, efficiency and user satisfaction [CL99, AKSS03] . The variety of definitions concerning usability attributes led to the use of different terms and labels for the same usability characteristics, or different terms for similar characteristics, without full consistency across these standards; in general, the situation in the literature is similar. For example, learnability is defined in ISO 9241-11 as a simple attribute, *‘time of learning‘*, whereas ISO 9126 defines it as including several attributes such as *‘comprehensible input and output, instructions readiness, messages readiness‘* [AKSS03].
As a basis for our following up survey we underlay a usability definition for modeling languages in model development and model interpretation scenarios including attributes as follows:
The usability of modeling languages is specified by learnability, memorability, effectiveness, efficiency, user satisfaction and perceptibility. The learnability of modeling languages describes the capability of a modeling language to enable the user to learn ap-

plying models based on particular language. The modeling language and its semantics, syntax and elements should be easy to remember, so that a user is able to return to the language after some period of non-use without having to learn the language and especially the application of models developed with specific language again. Effective model application should be supported by particular language for reaching a successful task accomplishment. Modeling languages should be efficient to use, so that a high level of working productivity is possible. Users have to be satisfied when using the language. For model interpretation scenarios the language should offer a convenient perceptibility regarding structure, overview, elements and shapes so that a user is able to search, extract and process available model information in an easy way.

3 Model of Hypotheses

In the following we show our hypotheses supported by theory. The motivation for those hypotheses lies in the general assumption that language-based metaproperties such as complexity (no. of elements, relations and properties) and visual properties (no. of various shapes and colors) influence the usability of graphical modeling languages [SCR11].

HYPOTHESIS1 Process Modeling Languages are more usable than System Development Languages

In general, usability means focusing on users and especially behavioral aspects [Nie93]. We hypothesize that languages applied for business process modeling result in higher usability values than languages applied for developing the conceptual design of application systems. Theoretical foundation of this hypothesis can be found in neuroscience and psychological research. Neuroscience research found out that typically, subjects show a processing advantage for concrete concepts over abstract concepts [CW05]. The reason for that lies in the fact that abstract concepts lacking the direct sensory referents of concrete concepts [Pai86]. Additionally, subjects have a greater availability of contextual information in the knowledge base for concrete concepts [SS83]. We deduce that users of process modeling languages have greater availability of contextual information due to concrete imagination of tasks and similarities to familiar domains.
Additionally, we introduce further theoretic background from cognitive load theory. The intrinsic cognitive load is determined by information complexity and interaction [PMB10]. For example, learning elements of modeling languages results in a low intrinsic cognitive load. In this case, the difficulty of learning a language and consequently the intrinsic cognitive load is strongly connected with the range of elements a language consists of. Contrariwise, the element interaction by means of syntactical and semantic element relations leads to a high intrinsic cognitive load. Consequently, the syntactical complexity of modeling languages highly determines the cognitive load. For example, a class diagram of the UML consists of various different relations such as generalization, aggregation, composition etc. set between classes. Due to this, the element interaction is much higher compared to control flows and object flows in process modeling languages.

We state, that this issue determines the usability of graphical modeling languages and subsequently leads to different usability values between process and system development languages.

HYPOTHESIS2 Behavioral Languages are more usable than Structural Languages

In general, this hypothesis builds on H1. The Object Management Group (OMG) structures the languages of the UML in (a) behavioral and (2) structural languages [OMG05]. H2 assumes that users are more familiar with behavioral languages such as UML Activity Diagram or UML Use Case Diagram due to concrete imagination of tasks to be modeled or interpreted compared to structural diagrams such as UML Class Diagram. We presume that this fact leads to different usability values between behavioral and structural languages.

HYPOTHESIS3 Complex Languages are not as usable as simple Structured Languages

This hypothesis is deduced from the observation that humans have limited cognitive capacity [SW07, GW04]. We presume that more complex languages are harder to absorb and understand in the humans brain than models with less complexity [RD07]. The rationale for this observation is quite obvious. Consequently, we assume that languages with high complexity result in low usability values compared to languages offering low complexity.

HYPOTHESIS4 Languages offering high visual differentiation are more usable than languages offering low visual differentiation

In the modeling language domain visual differentiation is strongly connected with the number of different element colors and geometric element shapes set in the specification of the modeling language [MH10]. Hall and Hanna (2004) analyzed the impact of color on web usability attributes in an empirical survey. They concluded that the application of different colors results in a higher grade of website structuredness, which leads to more efficient information processing in the users brain [HH04].
Moody and Heymans (2010) found out that visual differentiation of language properties impact cognitive effectiveness in practical usage scenarios [MH10]. Transferring this leads to the assumption that more element colors set in the languages metamodel lead to more information structuredness, which supports the usability of modeling languages.
Furthermore we assume that the variance of different geometric shapes depicting different element types is positively influencing information processing in the users brain. The theoretical basis for this assumption is initially given by Comber and Maltby (1997). They concluded that screen complexity including the application of various geometric shapes is a positive influencing variable of usability [CM97].
Consequently, all these theoretical findings let us hypothesize, that languages offering high visual differentiation are more usable than languages offering low visual differentiation.

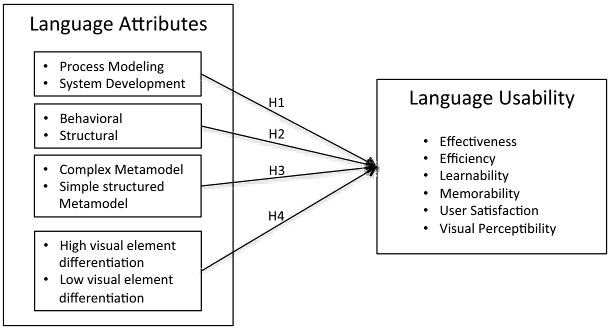

Figure 1: Model of Hypotheses

4 Measurement Items

The general usability measurement method in this paper is based on a Famework for Usability Evaluation of Modeling Languages (FUEML) [SRC10]. FUEML proposes the differentiation of evaluation procedure in usability attributes specified for each model development and model interpretation scenarios. Furthermore, the framework contains metrics for measuring and comparing the resulting values germane to all graphical modeling languages. In the following each attribute is specified and adapted on the domain of graphical modeling languages.

EFFECTIVENESS
Evaluating effectiveness requires analysis of task output with measuring quantity and quality of goal achievement [RMB+93]. Quantity is defined as the proportion of task goals represented in the output of a task. Quality is the degree to which the task goals represented in the output have been achieved [BM94]. Bevan (1995) defined effectiveness as a product of quantity and quality [Bev95]. Transferring this to our model, indicating manifest variables for measuring effectiveness are the grade of completeness and the grade of correctness of a model interpretation and model development task.

$$Effectiveness\,(F) = \frac{\sum_{i=1}^{n}\left(N_{task,i} + E_{task,i}\right)}{\sum_{i=1}^{n}\left(N_{goals,i} + E_{goals,i}\right)} * \frac{\sum_{i=1}^{n}\left(N_{goals,i} + E_{goals,i} - R_i\right)}{\sum_{i=1}^{n}\left(N_{goals,i} + E_{goals,i}\right)} \tag{1}$$

EFFICIENCY
The efficiency is the amount of human, economical and temporal resources. Measures of efficiency relate to the level of effectiveness achieved to the expenditure of resources. Measure values of efficiency include time taken to complete tasks, i.e. duration time T for performing a model interpretation and model development tasks.

$$Efficiency\,(G) = \frac{F}{T} \tag{2}$$

LEARNABILITY
Learnability describes the ease of learning the application (i.e. interpretation) of modeling languages. For this characteristic, the standard measure values are based on task completion rates and the task accuracy [SDKP06]. Hence, learnability can be measured by the rate of difference when the user repeats evaluation sessions [Bev95]. Nielsen (2006) insists that highly learnable systems could be categorized as "allowing users to reach a reasonable level of usage proficiency ..."[Nie93]. Furthermore, Nielsen (2006) proposes measuring proficiency by quantity and quality and of task fulfillment. Thus, we chose grade of completeness and grade of correctness as basic variables for measuring learnability. With conducting two measuring points mp and mp+1, it is possible to analyze the relative difference between mp and mp+1 for indicating learnability, i.e. individual learning progress in percent.

$$\Delta Learnability = \frac{G_{mp+1} - G_{mp}}{G_{mp}} \tag{3}$$

VISUAL PERCEPTIBILITY
The visual perceptibility focusing on model interpretation scenarios is measured by using the method of eye tracking with analyzing the users visual attention [Gor04]. In our research we aim to include eye tracking for measuring users cognitive processes i.e. information search and information extraction during model interpretation process. Concerning an eye tracking experiment for evaluating the visual perceptibility of modeling languages a large number of fixations implies an intensive search to explore the models diagram structure. This fact complicates the interpretation of a model. Furthermore, we aim to analyze the difficulty of information extraction in a model. Byrne et al. 1999 propose tracking fixation duration time as a measure for information extraction [BADM99]. From this follows that longer fixations times during an interpretation process are indicating a participants difficulty extracting information from a model.

USER SATISFACTION
Compared to the other latent variables in our research model, the individual satisfaction of a user while developing or interpreting a model is a user subjective criterion that can be measured best by using standardized questionnaires [KC93]. Currently no standardized method for measuring user satisfaction in the modeling domain exists. Therefore, we mapped questionnaires focusing on system and website usability [KC93, AGDD05]. For evaluating user satisfaction we developed a questionnaire, which consists of thirty items structured in

1. general impression,
2. satisfaction in development scenarios and
3. satisfaction in interpretation scenarios.

We measured the constructs with 5-point Likert-scales. The development of this questionnaire is generally contributing to the Questionnaire for User Interaction Satisfaction

(QUIS) and additionally the Software Usability Measurement Inventory [CDN88, KC93].

MEMORABILITY
Memorability is best measured as proficiency after a period of non-use provided a user has already learned a language. The non-use period can be minutes for simple element meanings, hours for simple syntactic regulations and days or weeks for measuring a complete modeling language [SDKP06]. Accordingly, the measure values for memorability are neglect curves and time-delayed knowledge tests. Concerning the usability of modeling languages, the user must remember the different elements and its intended meaning (semantics), the syntax and the application. In due consideration of Nielsen 2006, the measuring points interval should be several weeks regarding memorability [Nie93]. Thus, for measuring memorability we decided to use a knowledge test consisting of items focusing on elements and relations, syntax and the application of particular language.

LANGUAGE COMPLEXITY
Referring to Rossi and Brinkkemper (1996) elements, relations and properties can be abstracted and defined as modeling language complexity. The language complexity influences the usability attributes. For analyzing the languages complexity Welke (1992) and additionally Rossi and Brinkkemper (1996) developed metrics based on the OPRR data model. Transferring this to our approach metrics such as the number of object types (i.e. class), number of relationship types (i.e. association) and the number of property types (i.e. class name) are relevant for analyzing the complexity of a modeling language. The following metric was developed under consideration of Rossi and Brinkkemper (1996) and defines a complexity vector inside of a three- dimensional coordinate system:

$$C(M) = \sqrt{O^2 + R^2 + P^2} \tag{4}$$

COVARIATES
We analyzed two moderating variables, which affect causal relations in our model: First, the participants experience of developing or interpreting models and secondly the particular complexities of development or interpretation tasks influence the causal relations in our study. For measuring modeling experience we track participants individual experience in 1) general modeling experience and 2) language experience on a 5-point Likert-scale. Finally, we operationalized model complexity by three indicator-variables: number of elements and relations (size), connectivity degree and semantic spread. With running statistical analysis we include defined variables as covariates.

5 Research Design

This study used a large set of various data collection methods for measuring different usability variables. Furthermore, we introduced two data collection sessions per modeling language. The data collection focused on model development and model interpretation tasks. Within these sessions we collected error rates, grades of completeness and task finishing times for calculating efficiency, effectiveness and learnability, which is the relative learning growth between two data collection sessions. Additionally, we introduced the method of eye tracking for analyzing visual perceptibility of modeling languages specifically for the model interpretation scenario. The instruments were either adapted from traditional usability research or we developed new measurement items focusing on modeling languages. A pretest was conducted prior collecting data for the field test. All pilot test participants were excluded from the analysis sample.

DATA COLLECTION

The data collection was based on two different modeling concepts and connected languages. On the one hand process based languages, Event driven Process Chains (EPC), UML Activity Diagrams and on the other hand structure based modeling languages, UML Use Case and UML Class Diagrams. We chose these languages due to their cross-variability concerning the context of use, modeling concept, language complexity and visual properties. For measuring learnability we introduced a second measuring point. The first part of

	Context of Use	Modeling Concept	Language Complexity	Degree of different Geometrics*	Degree of different Colors**
EPC	Business Process Modeling	behavioral	19.26	1.00	0.80
UML Activity Diagrams	Software Engineering	behavioral	11.18	0.75	0.25
UML Class Diagrams	Software Engineering	structural	26.40	0.86	0.29
UML Use Case Diagrams	Software Engineering	behavioral	10.39	0.33	0.33

* Number of different shapes/total number of different elements
** Number of different element colors/total number of different elements

Table 1: Language Properties

each session focused on model development based on a given scenario described textually. For model development we used the modeling tools Bflow*-toolbox and ArgoUML. In the second part the students were confronted with the interpretation of given models. The interpretation scenario was structured in two parts. The first part was focusing on general observation while the second part included verbal interpretation of given models. Data was collected by using the Tobii Eye Tracker T60. The model development and interpretation tasks were based on common models and domain descriptions related to the modelling languages used in this survey.

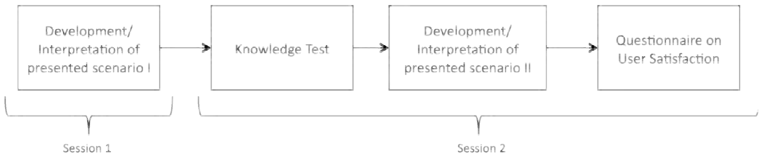

Figure 2: Survey Design

PARTICIPANTS
The sample includes second year students of business informatics. The experimental data collection, the questionnaire and the knowledge test were conducted with these students. The overall sample size amounts 114 students, 47% female and 53% male. Even though a choice of students for experiments has sometimes been criticized for lack of external validity, we agree with Gemino and Wand (2004) as well as Batra and Davis (1992) [GW04, VAKV08]. They confirmed that the selection of students over practitioners could in fact be advisable. Participants that are able to bring to bear prior knowledge in software and process engineering are sufficient for analyzing human impact criteria in the domain of graphical modeling languages [SDKP06] . Hence, the selection of students overcomes the problem of controlling for any bias in technique or domain familiarity.

6 Data Analysis and Results

The testing of our hypotheses was completed using an analysis of covariance (ANCOVA) technique. In our study ANCOVA was used to determine whether there is a statistically significant difference between EPCs, UML Activity Diagrams, UML Class Diagrams and UML Use-Case Diagrams regarding their impact on defined usability attributes considering user experience and task complexity as covariates. The ANCOVA including the analysis of the adjusted means were conducted using SPSS 19. All variables were tested for normal distribution and consequently validated using Kolmogorov-Smirnov z-values. In table 2 we present an aggregation of our results calculated applying the metrics of the applied evaluation framework shown in section 3. The values for User Satisfaction and Memorability were normalized between 0 and 1 and are based on the arithmetic mean of particular questionnaire and knowledge test items.

HYPOTHESIS H1 IS MOSTLY CONFIRMED
H1 hypothesized that business process modeling languages are more usable than system development modeling languages. This hypothesis is supported by our empirical results mostly. As well in development as in interpretation scenarios, EPCs are the most effective modeling language in our survey. In our survey, UML Activity Diagrams are most efficient in development and interpretation scenarios. Thus, this is not contributing to H1, since an Activity Diagram is a specific UML-language for depicting application system processes. However, the fact that EPCs are more efficient than UML Use Case and UML Class Diagrams is supporting this hypothesis. In most cases, the learnability of EPCs in model development scenarios is higher compared to the other modeling languages analyzed in this survey. In model interpretation scenarios the learnability is best with UML Use Case Diagrams. Concerning the memorability of different language properties and the particular application, UML Use Case Diagrams fits best. Users are most satisfied with applying EPCs in model development and model interpretation scenarios. Furthermore, searching for information is most efficient in EPC diagrams. However, the extraction of information is obviously most efficient in UML Class Diagrams. We conclude that H1 is somewhat supported for model interpretation scenarios. However, H1 is strongly supported for model

		EPC	UML Activity Diagram	UML Use Case Diagram	UML Class Diagram
Language Complexity		19.26	11.18	10.39	26.40
Visual Differentiation	Geometrics* Colors**	1.00 0.80	0.75 0.25	0.33 0.33	0.86 0.29
Effectiveness (=Grade of completeness*Grade of correctness)	Development (p=0.007), (F=4.231)	**0.943**	0.897	0.912	0.836
	Interpretation (p=0.000), (F=7.463)	**0.983**	0.970	0.891	0.882
Efficiency (=Effectiveness/Task time)	Development (p=0.071) (F=2.405)	0.074	**0.099**	0.051	0.033
	Interpretation (p=0.046) (F=2.466)	0.411	**0.438**	0.382	0.321
Learnability (=Rise of Efficiency in% between Session 1 and Session 2)	Development (p=0.042), (F=5.236)	**121.134**	13.549	45.576	116.469
	Interpretation (p=0.024), (F=6.563)	45.267	0.294	**63.321**	-8.422
Memorability (Normalized between 0 and 1)	Both Development and Interpretation (p=0.000), (F=18.255)	0.804	0.600	**0.815**	0.496
User Satisfaction (Normalized between 0 and 1)	General Impression (p=0.000), (F=8.947)	**0.734**	0.679	0.649	0.577
	Development (p=0.017), (F=3.540)	**0.748**	0.717	0.704	0.614
	Interpretation (p=0.037) (F=4.789)	**0.786**	0.777	0.749	0.734
Perceptibility (Information search=Fixation count) (Information extraction=fixation Duration)	Interpretation (p=0.000), (F=19.971) Information Search	**111.000**	188.000	153.000	248.000
	Interpretation (p=0.001), (F=8.768) Information Extraction	97.920	74.280	84.570	**73.870**

Note. All shown values are significant at least at the 0.05 level. Bold values show best language results for each usability attribute.

* Number of different shapes/total number of different elements

** Number of different element colors/total number of different elements

Table 2: Results

development scenarios. Overall, we conclude that H1 is mostly supported for model development scenarios by our results.

HYPOTHESIS H2 IS STRONGLY CONFIRMED
H2 hypothesized that behavioral languages are more usable than structural languages. Considering our results, we conclude that H2 is fully supported except the information extraction variable in interpretation scenarios.

HYPOTHESIS H3 IS PARTLY CONFIRMED
H3 assumes that complex languages are not as usable as simple structured languages. This hypothesis is partly confirmed by our results. For example, class diagrams (complexity=26.40) are comparable complex and our results show that they are not as usable as EPCs (complexity=19.26), UML Activity Diagrams (complexity=11.18) and UML Use Case Diagrams (complexity=10.39). So far, this fact supports our hypothesis. However, EPCs are more usable than UML Activity Diagrams and UML Use Case Diagrams. Although, they consist of more elements, relations and properties i.e. they are more complex.

HYPOTHESIS H4 IS FULLY CONFIRMED
H4 presumes that languages, which offer a high visual differentiation, are more usable than languages offering low visual differentiation. This hypothesis is confirmed by our empirical results. EPCs, which offer the highest visual differentiation in the metamodel, are the most usable language in our survey.

7 Discussion

With the exception of H4, our results for H1-H3 are indeed surprising because we cannot confirm or delete a hypothesis with 100 percent. As shown in Table 1 every language has advantages in usability results on the attribute level. We conclude that it is not possible to calculate a single usability measure for evaluating modeling languages. Our results show that different languages have different advantages and disadvantages on the usability attribute level. This finding is contributing to conclusions of Birkmeier et al. (2010). They conducted a survey including EPC and UML Activity Diagrams on selected usability measures. Their general conclusion was that it is not possible to make general recommendations [BKO10].
Our results confirm the strict separation between model development and model interpretation scenarios proposed in the usability evaluation framework (FUEML) underlying the measurement model in our survey. For example, UML Class Diagrams, and EPCS have a comparable learnability in model development scenarios. Contrariwise, the learnability of these languages is significantly less in model interpretation scenarios.
For almost all usability attributes the results are headed by behavioral languages. Considering this, we confirm that behavioral modeling languages result in higher measuring values concerning the different usability attributes than structural modeling languages. Obviously, human beings show a neuro-processing advantage for concrete concepts over

abstract concepts [CW05]. Transferring this into the domain of graphical modeling languages leads to the conclusion that users of behavioral modeling languages have greater availability of contextual information (i.e. concrete imagination, similarities etc.) compared to structural modeling languages. This is extended by the fact, that EPCs show high usability in many attributes. We deduce, that process modeling is more usable than system modeling due to different abstraction levels.

Additionally, our results show that languages offering low complexity result in high values indicating memorability. This finding is contributing to Kintsch (1998), who shows that cognitive processes underlie the comprehension of a specific domain [Kin98]. Nembhard and Napassavong (2002) found out that the complexity of a specific domain influences memorability negatively [NN02]. Since our results contribute to the intrinsic Cognitive Load (see section 2), this fact underlines the application of Cognitive Load Theory in usability assessment of graphical modeling languages.

Our results indicate that languages offering visual differentiation in the metamodel are generally more usable than languages offering low visual differentiation. These results underline findings of Moody and Hillegersberg (2009). They found out, that increasing visual differentiation in language specifications could optimize the visual effectiveness and consequently the visual perceptibility of graphical modeling languages [MH09] .

The search for information in diagrams developed by using specific languages is most efficient in EPC diagrams. A probable reason for this may be due to the fact that the specification and the metamodel of EPCs strictly set the use of different colors and different shapes for particular elements. Accordingly, the information structuredness in EPC diagrams is higher compared to UML Activity, UML Use Case and UML Class Diagrams leading to more efficient information search procedures in model interpretation scenarios [MH09].

Contrariwise, the extraction of information out of class diagrams is easier compared to EPCs, UML Activity and UML Use Case Diagrams. However, at this stage it is essential to consider the quality of information extraction. With comparing the values for effectiveness we conclude that information extraction out of class diagrams is comparable easy but the quality of information extraction is comparable low. A possible reason for that lies in the fact that the size of UML Class Diagrams is comparable low in relation to information density. This increases the efficiency of information extraction. Indeed, the information density in UML Class Diagrams is comparable higher than the density of information in EPCs, UML Activity Diagrams and UML Use Case Diagrams. This is based on the fact that UML Class Diagrams consist of comparable less elements and relations. But various properties such as multiplicities can be added to them resulting in a higher grade of variation. This fact lets class diagrams appear more difficult to use than EPCs, UML Activity Diagrams and UML Use Case Diagrams.

Furthermore, our results imply that users prefer process modeling languages for modeling and interpreting scenarios. On the one hand we suppose that this behavior is based on the fact that process modeling languages are comparable less complex and easy to understand due to greater availability of contextual and concrete information [Pai86, SS83].

Another interesting finding is that UML Activity Diagrams support most efficient task accomplishment in our survey in model development as well as model interpretation scenarios. A possible reason might be that they are not as complex as EPCs. For example, UML Activity Diagrams include the swimlane concept for adding information about specific task (=activity) responsibilities whereas EPCs need the organizational unit for every task (=function). This increases the visual spread of an EPC-diagram significantly.

8 Implications

This paper explored usability-driven differences of graphical modeling languages using the FUEML framework. The implications that can be deduced from our results give insights into modeling in education, modeling language development and business process modeling.

MODELING IN EDUCATION
Within the scope of this study we could mostly confirm that business process modeling languages are more usable than system development languages. What does that imply for educational usage of modeling languages? We recommend that training courses on modeling languages for software development should be more intensive than courses on business process modeling languages. This recommendation focuses on the pure training with the modeling language with the exception of tool and domain influences. Our results confirm that different modeling concepts have different impact on the usability and consisting attributes. Furthermore, our results show that behavioral languages are more usable than structural languages.
Again, this implies that structural modeling languages such as the UML Class Diagram need more educational language-based training intensity than behavioral languages.

MODELING LANGUAGE DEVELOPMENT
An interesting implication could be deduced concerning the use of EPCs or UML Activity Diagrams for modeling real-world scenarios. EPCs are more usable than UML Activity Diagrams. We trace this result back on the fact that EPCs support visual differentiation due to the use of various colors and geometric shapes in the language specification. Consequently, the usability-oriented optimization of UML might be reached by adding different colors and different shapes. This implication coincides with findings of Moody and Heymans (2010) [MH10].
Additionally, our results imply that UML Activity Diagrams support more efficient task accomplishments in model development and model interpretation scenarios. Obviously, activity diagrams allow users fast model development and model interpretation procedures. A possible reason is the low language complexity of Activity diagrams compared to EPCs. It seems that reducing complexity and raising visual differentiation support the usability of graphical modeling languages. We are aware that complexity reduction of modeling languages possibly may impact the expense of explanatory. Transferring this, we conclude that usability issues are only partly considered in the new BPMN 2.0 release of the Object Management Group in January 2011.
For example, with this new release the BPMN was extended with plenty of technical extensions. However, additional visual differentiations (i.e. colors, geometric shapes etc.) were not added yet. Looking at another major language, the UML 2.0, the situation is

quite equivalent. It seems, that language specification organizations focus on technical language optimization whereas the human being acting as user is partly ignored in those optimizing activities. Our empirical study shows that it could be worth thinking about usability-related topics in further development of graphical modeling languages.

BUSINESS PROCESS MODELING
The results show that EPCs offer the best usability in our survey. In companies the importance of business process modeling has steadily risen. Consequently, the development and interpretation of models become an issue of organizational concerns. How efficiently can models be developed or interpreted? For the model interpretation scenario, questions such as whether employees understand the information modeled do appear.
Thus, companies aiming for fast, complete and correct model interpretation, e.g. business process consulting companies, typically apply modeling languages offering high variability in visual properties. In many cases those companies customize languages such as the BPMN by adding colors or shapes to support complete and accurate model interpretation. Since EPCs offer highest usability values in development and interpretation scenarios, our results support this course of action.

9 Further Research

In this paper we conducted a comparison of different selected modeling languages and their impact of usability attributes. What we dont know yet, are the characteristics impacting usability in the domain of graphical modeling languages specifically. This could be very interesting for defining recommendations focusing on further development of existing modeling languages. Thus, we claim that a further inductive, causal study might bring out new and important findings in that research area.

References

[AGDD05] D. Armstrong, G. Gogarty, D. Dingsdag, and J. Dimbley. Validation of a Computer User Satisfaction Questionnaire Validation of a Computer User Satisfaction Questionnaire to Measure IS Success in Small Business. *Journal of Research and Practice in Information Technology*, 37(1):22–38, 2005.

[AKSS03] A. Abran, A. Khelifi, W. Suryn, and A. Seffah. Consolidating the ISO Usability Models, 2003.

[BADM99] Michael D. Byrne, J.R. Anderson, Scott Douglass, and Michael Matessa. Eye Tracking the Visual Search of Click-Down Menus. In *Proceedings of CHI'99*, pages 402–409, 1999.

[Bev95] Nigel Bevan. Measuring usability as quality of use. *Software Quality Journal*, 4:115–150, 1995.

[BKO10] Dominik Birkmeier, Sebastian Klöckner, and Sven Overhage. An empirical comparison ot the usability of BPMN and UML Activity diagrams for business users. In *Proceed-*

ings of the 18th European Conference on Information Systems (ECIS), South Africa, 2010.

[BM94] Nigel Bevan and Miles Macleod. Usability Measurment in Context. *Behaviour and Information Technology*, 13(1):132–145, 1994.

[CDN88] John Chin, Virginia Diehl, and Kent Norman. Development of an instrument measuring user satisfaction of the human-computer interface. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–218, 1988.

[CK06] Constantinos Coursaris and Dan Kim. A Qualitative Review of Empirical Mobile Usability Studies. In *Proceedings of the 12th Americas International Conference on Information Systems (AMCIS)*, pages 85–94, 2006.

[CL99] L.L. Constantine and L.A.D. Lockwood. *Software for Use: A practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley, New York, 1999.

[CM97] T. Comber and Jr Maltby. Layout complexity: does it measure usability?, 1997.

[CW05] Sebastian Crutch and Elizabeth Warrington. Abstract and concrete concepts have structurally different representational frameworks. *Brain*, 128(1):615–627, 2005.

[DR99] J. Dumas and J. Redish. *A practical guide to usability testing*. Greenwood Publishing Group, Westport, 2nd edition edition, 1999.

[Gor04] Ian E. Gordon. *Theories of visual perception*. Psychology Press, Hove, 3. ed. edition, 2004.

[GW04] Andrew Gemino and Yair Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4):248–260, 2004.

[HH04] Richard Hall and Patrick Hanna. The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention. *Behaviour and Information Technology*, 23(3):183–195, 2004.

[IEE90] IEEE610.12-1990. Standard Glossary of Software Engineering Terminology. Technical report, Institute of Electrical and Electronics Engineers, 1990.

[ISO98] ISO/IEC9241-11. Ergonomic Requirements for Office Work with visual Display Terminals (VDTs); Part 11: Guidance on Usability. Technical report, International Organization for Standardization, 1998.

[ISO06] ISO/IEC9241-110. Ergonomics of Human-System-Interaction; Part 110: Dialogue Principles. Technical report, International Organization for Standardization, 2006.

[KC93] J. Kirakowski and M. Corbett. SUMI: The Software Usability Measurement Inventory. *British Journal of Educational Technology*, 24(1):210–212, 1993.

[Kin98] Walter Kintsch. *Comprehension: A Paradigm for Cognition*. Cambridge University Press, Cambridge, Melbourne, 1998.

[Lud03] Jochen Ludewig. Models in software engineering - an introduction. *Software and Systems Modeling*, 2(1):5–14, 2003.

[MH09] Daniel L. Moody and Jos Hillegersberg. Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML Family of diagrams. *Lecture Notes in Computer Science*, 5452(1):16–34, 2009.

[MH10] Daniel L. Moody and P. Heymans. Visual Syntax does matter: improving the cognitive effectiveness of the i* visual notation. *Requirements Engineering*, 15(1):141–175, 2010.

[Nie93] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1st edition, 1993.

[NN02] D. Nembhard and O. Napassavong. Task complexity effects on between-individual learning/forgetting variability. *International Journal of Industrial Ergonomics*, 29(2):297–306, 2002.

[OMG05] OMG. Introduction to OMG's Unified Modeling Language™ (UML®). Technical report, Object Management Group, 2005.

[Pai86] A. Paivio. *Mental representations: a dual coding approach*. Oxford University Press, Oxford, 1986.

[PMB10] J. Plass, R. Moreno, and R. Brünken. *Cognititve Load Theory*. Cambridge University Press, 2010.

[PRS+94] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. *Human Computer Interaction*. Addison-Wesley, Wokingham, 1994.

[RD07] Jan C. Recker and Alexander Dreiling. Does it matter which process modelling language we teach or use? An experimental study on understanding process modelling languages without formal education. In *Proceedings of 18th Australasian Conference on Information System (ACIS 2007)*. University of Southern Queensland, 2007.

[RMB+93] R. Rengger, M. Macleod, R. Bowden, M. Blaney, and N. Bevan. *MUSiC Performance Measurement Handbook*. National Physical Laboratory, Teddington, UK, 1993.

[SCR11] Christian Schalles, John Creagh, and Michael Rebstock. Usability of Modelling Languages for Model Interpretation: An Empirical Research Report. In *10th International Conference on Wirtschaftsinformatik (WI2011)*, Zurich, 2011.

[SDKP06] Ahmed Seffah, Mohammad Donyaee, Rex Kline, and Harkirat Padda. Usability measurement and metrics: A consolidated model. *Software Quality Control*, 14(2):159–178, 2006.

[Sha91] B. Shackel. Usability - Context, framework, definition, design and evaluation. In B. Shackel and S. Richardson, editors, *Human Factors for Informatics Usability*, pages 21–38. University Press, Cambridge, 1991.

[SRC10] Christian Schalles, Michael Rebstock, and John Creagh. Ein generischer Ansatz zur Messung der Benutzerfreundlichkeit von Modellierungssprachen. In Gregor Engels, Dimitris Karagiannis, and Heinrich C. Mayr, editors, *Modellierung 2010*, volume 161 of *Lecture Notes in Informatics (LNI)*, pages 15–30, Klagenfurt, 2010. Gesellschaft für Informatik (GI).

[SS83] P.J. Schwanenflugel and E.J. Shoben. Differential context effects in the comprehension of abstract and concrete verbal materials. *Journal of Experimental Psychology*, 9(1):82–102, 1983.

[SW07] Keng Siau and Yuan Wang. Cognitive evaluation of information modeling methods. *Information and Software Technology*, 49(5):455–474, 2007.

[VAKV08] M. Vuolle, A. Aula, M. Kulju, and Tejia Vainio. Identifying Usability and Productivity Dimensions for Measuring the Success of Mobile Business Services. *Advances in Human-Computer Interaction*, 2008.