A Guideline to Test the Understandability of Notations

Susanne Patig[1]

[1] Otto-von-Guericke-University of Magdeburg, FIN-ITI, Universitätsplatz 2,
D-39106 Magdeburg, Germany
E-mail: patig@iti.cs.uni-magdeburg.de

Abstract. Which of several notations is easier to understand? This question arises for conceptual models, query or programming languages. Answering it empirically by experiments involves numerous decisions (e.g., experimental design, dependent variables and sample size). This paper presents a methodical guideline for planning, conducting and evaluating such experiments. The suggestions are supported by the open-source tool `notate`. Applying the guideline and the tool in an experiment revealed that a graphical notation used within SAP AG is precise as a textual notation, but is also more difficult to understand.

Keywords: Understandability, Conceptual Models, Experimental Design

1 Practical Motivation

Ideally, software development starts from abstract descriptions of use cases, database schemas, system architecture, or messages exchanged between software components. Usually, the descriptions follow a notation provided by some model (such as the entity-relationship (ER) model) or modelling language (such as the unified modelling language UML). A *notation* consists of constructs (geometric figures, characters, numbers, or symbols) and a syntax, which restricts the allowed construct combinations.

Notations are rather arbitrary (for ER models see, e.g., [15]), and new ones continuously spring up at major conferences like ER, ATPN and CAISE. But, inventing notations is not only an academic exercise - it can become necessary if ideal software development is done in praxis. SAP AG experienced this phenomon in developing its latest application system:

In an iterative process including several steps of approval, SAP AG first created formalized descriptions of the new system's architecture and functionality. The developers then had to adhere exactly to the approved system descriptions[1]; thus, that the notations used in the descriptions must be easy to understand and unambiguous.

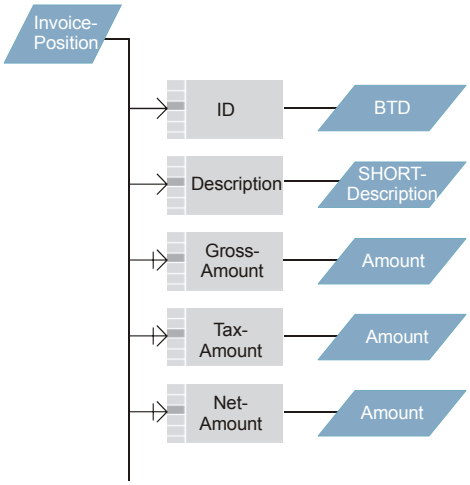
Since the architecture of the new application system differs from what has been known to date, it could not be expressed by common models without extending or modifying them. Hence, SAP AG decided to create a set of new models, each representing another view on the system. Among them, I encountered redundancy, since

[1] Although only smaller parts of the source code are automatically generated, SAP AG uses the term 'model-driven development'. The system descriptions are called 'models'.

submitted to: *ER 2007 - 26th Inter. Conference on Conceptual Modeling (ER 2007)*

complex data types are described simultaneously by two notations (see Fig. 1): The *graphical* one represents the attributes ('elements') of the complex data type by rectangles, their (element) data types by parallelograms, and uses special arcs to show whether or not the attributes are optional ('cardinality'[2]). The textual notation expresses the same information by tables, where attributes, their data types, and cardinalities are contained in rows[3].

a) Graphical notation



b) Textual notation

NDT	Name	Data Type	Card.
Expense-Report			
	ID	ID	1
	EmployeeID	ID	1
	Payment-Currency	Currency-Code	1
	Stay-Description	MEDIUM-Description	0..1
	StayLocation-Name	MEDIUM-Name	0..1

Fig. 1. Equivalent Notations for Complex Data Types (Content Extract, © SAP AG)

Many developers I asked preferred the table notation, but subjective judgements are too weak to change a decision that touches the work of hundreds of people. Finally, I was allowed to examine which of the notations was easier to understand.

Reviewing how understandability is usually tested in computer science raised many questions (see Section 2). To answer them, I buried myself in the basics of experimental research in psychology and assembled the guidelines found in Section 3. Section 4 reports the results I obtained by applying the guideline to test the understandability of the notations in Fig. 1. Finally, Section 5 highlights the contributions of this paper.

2 Research on Understandability in Computer Science

The debate about various styles of notations and their ease of use, including understandability, has a long history in computer science. One of the earliest disputes took place in artificial intelligence by praising the merits of either predicate logic [8], [12], which is usually[4] written as text, or visual representations and diagrams [13], [19]. I

[2] The term 'cardinality' is slightly misleading, since multi-valued attributes are not allowed.

[3] The SAP tables contain additional columns for value ranges, integrity conditions, etc. I omitted them in my experiment to keep the comparison fair.

[4] Conceptual graphs (http://conceptualgraphs.org/) are a graphically notated predicate logic.

Table 1. Experiments on the Understandability of Models or Languages

Study	Independent Variables (Levels)	Tasks (Number)	Dependent Variables	Experimental Design	N	Statistical Procedure	Results
[2]	Data model (EER, RDM)	Modelling (1 case)	Correctness (review), perceived ease of use	2 groups, matched in experience	42	t-test of means	• EER leads to higher correctness • No difference in perceived ease of use
[3]	Conceptual data model (EER, KOOM)	Modelling (1 case)	Correctness (review)	2 groups, random assignment	38	matched-pairs t-test for means	• Mostly no differences in correctness • Higher correctness of EER only for some facets
[5]	Graphical query languages	Comprehension (32), Query specification (14)	Correctness (review)	1 group, repeated measurement	27	χ^2-test on distribution	Graphical queries are: • Easy to comprehend • Not easy to specify
[9]	Database representation style (graphical, textual), database complexity	Query specification (20)	Correctness (review), solution time, perceived ease of use	2 x 2 factorial design, repeated measurement	36	3 way analyses of variance (ANOVA)	Compared to textual representations, graphical representations lead to: • Shorter time to specify a query • Higher correctness of the queries • Higher perceived ease of use.
[10]	Conceptual data models (EER, SOM, ORM, OMT)	Modelling (2 cases)	Correctness (review), modelling time, perceived ease of use	4 groups, random assignment	100	Duncan test	Increased correctness and faster solutions for EER and OMT
[14]	Conceptual models (DSD, ERM, OOM)	Comprehension (30)	Correctness, solution time	3 groups	121	ANOVA, correlation analysis	• Highest correctness for OOM • Shortest solution time for OOM, followed by DSD, ERM

Abbreviations: DSD: Data Structure Diagram, EER: Extended Entity-Relationship Model, (K) OOM: (Kroenkes) Object Oriented Model, N: Total number of participants, SOM: Semantic Object Model, ORM: Object Role Model, OMT: Object Modelling Technique, RDM: Relational Data Model

skip that discussion here, because all of the cited papers base their claims solely on examples - quite suggestive indeed, but only anecdotal evidence.

Empirical research on ease of use of, e.g., (conceptual) data models, query languages, or representation styles, relies on experiments. The list in Table 1 is not intended to be exhaustive, but merely to illustrate the variety of ways available to conduct such experiments.

An *experiment* is a scientific investigation in which one or more independent variables are systematically manipulated to observe their effects on one or more dependent variables [17]. The *independent variable* is the one whose effect is to be examined (e.g., 'data model'); the values it can take during manipulation are called *levels* (e.g., ERM, RDM) [17]. In contrast to the independent variable (which is given by the research question), the *dependent variable* can be chosen freely, provided that it serves as a measure of the effect [16]. In Table 1, the effect is ease of use.

The dependent variables, as well as the number of independent variables and the number of their levels, the number of participants, and the statistical procedure, all vary among the studies in Table 1. All these decisions relate to *experimental design*, i.e., the way participants are selected and assigned to experimental conditions [16].

The results of the experiments in Table 1 are not uniform. For example, the higher perceived ease of use of graphical notations as compared to textual ones [9] was not confirmed by the results in [2]. Such comparisons are to some extent unfair, since they presuppose a common definition of graphical and textual notations, which actually does not exist (see the comprehensive discussion in [18]). I discriminate as follows (analogous to [13]): The constructs of *textual notations* are (strings of) signs (characters, numbers, or symbols), which are allowed to form sequences or arrays. In contrast, the constructs of *graphical notations* are geometric figures (e.g., circles, rectangles, or arcs), which can be arbitrarily connected. Hence, ER diagrams and Fig. 1a) arise from graphical notations, whereas RDM schemas and Fig. 1b) stem from textual ones.

Even for graphical notations in Table 1, the results are contradictory. For example, [14] discovered a significantly higher correctness for object-oriented modelling than for ER modelling, whereas [3] reported the contrary, at least for some aspects. Is the result obtained by the larger number of participants more trustworthy? Does a more sophisticated statistical procedure guarantee greater validity? These questions are discussed in Section 3.4, which deals with the evaluation of experiments.

Table 1 does not help one in planning experiments, as it leaves the appropriate experimental design, task, dependent variable(s), etc. undecided. The guideline makes respective suggestions in Section 3.2; they are derived from the nature of experiments (Section 3.1), which must be regarded in conducting experiments (Section 3.3).

3 A Guideline for Experiments on Understandability in Computer Science

3.1 The Nature of Experiments

Experiments are always guided by the hypothesis that the independent variable(s) will *cause* the changes in the dependent variable(s) [17]. The amount of change of the de-

submitted to: *ER 2007 - 26th Inter. Conference on Conceptual Modeling (ER 2007)*

pendent variable *actually due* to the manipulation of the independent variable is called *primary variance* [16]. Unfortunately, primary variance cannot be obtained directly; it is 'hidden' in the total variance of the experimental data, which also contains secondary and error variance. Both are unwanted though different in their impact on the experimental results. *Error variance* is a consequence of random effects (e.g., faulty measurements or individual differences among participants). It affects the experimental data fairly equally and decreases precision, but will not disturb the statistical conclusions [17]. *Secondary variance*, on the other hand, stems from factors other than the independent variables (referred to as secondary or extraneous variables) that systematically influence the dependent variable(s)) and thus confound the results [17]. The degree to which the variation of the dependent variable can be attributed to the independent variable (rather than to some other factor) is called *internal validity*; it decreases if extraneous variables exist [17].

To cope with secondary variance, the extraneous variables must first be identified. Often they can be gathered from relevant literature. For instance, [2], [3], and [9] recognize the influence of task complexity and user experience on understandability. Other extraneous *variables* with known effect on the performance of participants are:

1. Related to the experimental **situation**: The location (noise, room temperature), the time of day (e.g., experiments in the morning usually yield better results than ones after lunch), and the equipment (failures, calibration) [6].
2. Related to the **persons** involved in the experiment [16]:
   - Participants: Their age, sex, mental ability, experience, motivation (to complete all tasks or to boycott the experiment).
   - Experimenter: The ability to instruct participants, bias (expecting a particular outcome unconsciously distorts the experimenter's behaviour or data gathering).

1. Related to the **conduct** of the experiment:
   - Position effect: Performance depends on the timely distance of a task from the start of the experiment (e.g., fatigue, getting bored, learning) [11].
   - Carry-**over effect**: The performance achieved in some task depends on whether or not some other task has been done before [17].

If the extraneous variables of some experimental situation have been identified, they can be controlled by the techniques described in Section 3.2. Controlling secondary variance while simultaneously minimizing error variance will maximize primary variance, i.e., the effect a researcher is interested in becomes more obvious. This is the basic idea behind experimental design (see Section 3.2). In the following, I will focus on experiments that deal with the understandability of notations in computer science.

3.2 Planning an Experiment

Experiments determine whether a particular hypothesis is either true or false. Hence, the hypothesis must first be formulated. A *hypothesis* is understood as a testable statement of a potential relationship between two or more variables [16]. Testability requires the hypothesis to be formulated in terms that are manipulable (for the independent variable) and measurable (for the dependent variable).

In the context of this paper, two generic types of hypotheses can occur:

- *One-tailed*: A particular notation can be understood easier than another one.
- *Two-tailed*: Particular notations differ in their understandability.

The statistical test procedure often dictates the type of hypothesis. If not, a one-tailed hypothesis is only allowed to contrast two notations and it must be reasonably sure which of them is superior [17]. This will rarely be the case (see the conflicting results in Table 1). Using a one-tailed instead of a two-tailed hypothesis increases the risk of Type I errors (see Section 3.4).

Formulating the hypothesis is strongly connected to selecting independent and dependent variables. Here, the *independent variable* (IV) - the one whose effect is to be examined - is notation, and the particular notations to be compared (e.g., ERM, RDM or a) and b) of Fig. 1) form its levels. Since the dependent variable must be tested under at least two conditions, at least two levels of the independent variable are necessary [17]. This can be achieved by contrasting either two notations or a notation and a control group. In a *control group*, the independent variable is not manipulated; natural language is used instead of a notation[5].

Rarely more than four levels of an independent variable are investigated. Examining more than one independent variable requires a factorial design (see Table 2). More than three independent variables are not recommended, because this will render statistical analysis rather cumbersome and interpretation difficult [16].

The *dependent variable* (DV) is the one on which the effect of the independent variable is measured. Experiments on understandability belong to the field of (cognitive) psychology, where the dependent variable generally refers to behaviour. Common measures of behaviour – potential DVs - are the following [16]:

- *Frequency*, e.g., the number of correct answers or solved problems.
- *Response latency (or response time)*, which is concerned with how long it takes for a behaviour to be emitted, e.g., how quickly a participant reacts.
- *Response duration*, i.e., the length of time behaviour occurs (e.g., how long a participant deals with a task).
- *Selection*, e.g., which of several notations a participant uses for modelling or which of several answers he or she chooses.
- *Amplitude*, measuring the strength of response, e.g., the brain activities in performing a task.

Better understandability manifests itself by a greater number of correct answers, shorter response times, and less brain activity [1], [4]. Measuring amplitude is reserved for neuroscience. Multiple-choice questions about the content of descriptions are based on the measure ‘selection’ (of answers), but the experiments in Table 1 used only open questions. Frequency can be easily counted, but requires the experimenter to identify correct solutions [4]. This becomes difficult if the task (see Table 1) consists in modelling or query specification, because assessing correctness ex-post by reviews bears the danger of experimenter bias.

Time-based measures do not primarily consider correctness (they also accept wrong solutions), but they need additional equipment and are thus susceptible to measurement errors. Nevertheless, it has a long tradition in psychology to draw

[5] [5] wrongly compares the actual understandability of some query language not to a control group, but to a theoretical and, hence, rather idealistic performance.

conclusions concerning mental processes from the time spent dealing with a problem (solution time, see Table 1) [1], [11]. However, such conclusions are double-edged: A shorter solution time may indicate either an easier problem (a notation that is easier to understand) or a participant with a higher IQ [16].

Perceptions (as the perceived ease of use in Table 1) are not a measure of behaviour in psychology because they are not observable and can therefore be easily distorted by the participants. Whether participants apply one notation or another to model a system is obvious. However, this choice is not solely due to the notation's ease of use, but also to the expertise and training of the participant [7].

The *experimental task* restricts the suitable measure of behaviour. Psychology knows numerous task types [1], [16]. In the experiments discussed here, the task should allow assessing the *ease of use* of notations, which has two dimensions: Creating descriptions (modelling) or specifications and understanding them (reading) (Table 1). It is more reliable to conduct experiments with the focus on understanding descriptions, because only cognitive psychology provides at least a few (yet rather vague and partially conflicting) explanations [1]. Moreover, this focus relieves us of the difficulties in evaluating whether some results of modelling are correct.

In summary, I suggest investigating not the ease of use of notations, but rather their understandability. It can be inferred from the reading of *descriptions*, provided that they refer to equivalent content and differ only in notation. The particular measure forming the dependent variable should guarantee unambiguous and uniform data gathering (see Section 3.3). The number of dependent variables is not limited. Using more than one measure of behaviour increases the reliability of the findings [4].

The next step in planning an experiment consists in controlling secondary variance. This is achieved by tackling its source, the extraneous variables (EV); see Section 3.1. The following *control techniques* exist [16], [17]:

1. *Remove* the EV from the experimental situation, e.g., use a quiet room.
2. Make the EV an *independent* variable if it has a strong influence on the dependent variable and can be systematically varied (e.g., the complexity of tasks). This control technique increases the explanatory value of experiments and concurrently makes experimental design and statistical analysis more complicated.
3. Hold the EV *constant* if the experimental design should be kept simple and the EV cannot be removed. Constancy is often applied to methodological aspects of an experiment, e.g., all participants are instructed by the same person, tested on the same time of day, etc. It guarantees that all conditions are identical except for the manipulation of the independent variable.
4. *Randomize* an EV that cannot be removed when its influence is not known (e.g., differences between male and female participants in understanding notations); must be neutralized (e.g., position effects, carry-over effects) or should be equated (e.g., mental ability, experience). Randomization, the main principle of statistical sampling, should be applied whenever possible, since it increases the *external validity* of the experiment, that is, its ability to be generalized [16].

An *experimental design* can be regarded as a general plan for (types of) experiments that join independent variables and control techniques. Table 2 summarizes the main experimental designs; all have been used in computer science (see Table 1).

In a *between-subjects* (or n-groups) *design*, the participants are randomly assigned to several groups [6]. Each group is treated by only one level of the independent variable (in our context, a particular notation). Thus, carry-over effects between alternative notations cannot occur. However, especially in small samples (number of participants in a group $n_i < 15$, total number of participants $\Sigma\, n_i = N < 30$) randomization may lead to groups that are unequal concerning individual characteristics of the participants (e.g., IQ, experience); this can skew the experimental outcomes.

A *block design* (synonyms: paired/matched groups) avoids unequal groups. Here, at least one *matching factor* strongly connected with the dependent variable is chosen (e.g., experience), the participants are classified according to the levels of this factor (e.g., high, medium, low experience) and randomly assigned to groups, so that each factor level is represented in all groups by the same number of participants [17]. In other words, the matching factor is kept constant. The effort this design causes is only worth if the matching factors are highly correlated; detecting them is difficult.

Table 2. Summary of Experimental Designs and Statistical Test Procedures

Design	**Between-subjects**	**Within-subjects**	**Block (Matched)**	**Factorial**
No. of IV (levels)	1 (n)	1 (n)	1 (n)	m > 1 (n)
No. of groups	n	1	n	m × n
Pro:	No carry-over effects	• Simple • Small samples • Constancy of individual characteristics	• Precise • No carry-over effects • Individual differences balanced	Interactions between IV can be examined
Contra:	• Unequal groups possible • Large samples	• Carry-over effects • Experimenter bias	• Effort • Matching factor must exist	• Large samples • Difficult to interpret for m > 3
Statistical test procedures				
Metric DV	♦: independent t ∗: F-test, ANOVA	♦: paired t-test of means ∗: MANOVA		MANOVA
Ordinal DV	♦: Mann-Whitney U ∗: Kruskal-Wallis H	♦: Wilcoxon signed rank test (matched) ∗: Friedman's χ^2		-
Nominal DV	♦/∗: χ^2 contingency test	♦: Sign test, McNemar's test of change ∗: Cochran's Q-test		-
Sample Size ♣	1-t: n_i = 20 [50] 2-t: n_i = 25 [60]	1-t: N = 11 [26] 2-t: N = 15 [35]	see between-subjects	2-t only, m = 3: n_i = 20 [50]

♣ To detect a large [a medium] effect (see Table 3) with $(1-\beta) = 0.8$ and $\alpha = 0.05$.

The *within-subjects* (or repeated-measurements) *design* is appropriate to yield significant results with a small sample without demanding preparation [6]. Here, one group is tested for all levels of the independent variables, i.e., for all notations. In this setting, all extraneous variables related to the persons involved in the experiment remain constant. The disadvantages of this design consist in carry-over effects and experi-

menter bias: As a result of carry-over effects, participants may transfer solutions obtained by some notation (e.g., content they have understood) to tasks that use another notation. Moreover, the expectations of the experimenter may impair his or her instructions (e.g., the favourite notation is explained better), or tip the participants off to the research goal. Section 3.3 shows how to avoid these disadvantages.

If more than one independent variable is investigated, a *factorial design* must be applied [16]. Here, the groups are the result of combining each level of an independent variable (e.g., notation) with each level of another one (e.g., experience). Since each participant is assigned to only one group, a large sample is required. The more levels or independent variables that are considered, the more difficult become the planning and the statistical evaluation of the experiment.

The appropriate experimental design depends on practical considerations (participant availability, acceptable effort, etc.) and on theoretical ones such as the control technique and the research question. Experiments that answer the question, "Does the IV have an effect?" are called *exploratory*, whereas *analytical* experiments deal with the question, "How much of an effect does the IV have?" [16]. Due to potentially unequal groups, two-groups designs are restricted to exploratory experiments which involve a control group. Factorial designs allow exploratory and analytical investigations [16]. Within-subjects designs must be carefully conducted to yield valid analytical results. The question we are interested in is analytical, since differences in the understandability of notations are already known.

The last step in planning an experiment is determining the *sample size N*, i.e., the total number of participants in all groups. The rough recommendations in Table 2 are derived from the requirements of the statistical test procedures, the acceptable statistical errors and the desired effects; this is discussed briefly in Section 3.4.

3.3 Conducting an Experiment

The actual experiment must properly put the chosen design into action, focusing on minimizing error variance.

Section 3.2 suggested assessing the understandability of notations based on reading descriptions (diagrams, formal text). Reading is measured by questions on the content read [1]. The number of correct answers for each notation and the response time can then act as dependent variables; both measures are independent.

Experimenter bias in determining correctness cannot occur if the questions have multiple-choice form and allow only one ultimate solution [4]. The solutions chosen by the participants constitute an additional (but correlated) dependent variable. Multiple-choice questions with three answers reduce the possibility of answering correctly by chance and, at the same time, do not confound the response time with the time for reading a long list of answers. Answers should be formulated positively (negations distort the solution time [1]) and not obviously false. Moreover, the correct answers should be spread over the options participants can choose.

The within-subjects design suggests itself as a simple one that requires few participants. External validity is achieved by assembling the group as heterogeneously as possible. Internal validity requires controlling the extraneous variables, which is quite easy in experiments on the understandability of notations. Carry-over effects are neu-

tralized if tasks that refer to distinct notations are presented in random order and if the content described in the tasks is comparable but not identical. To reduce the danger of experimenter bias, not only the notations of interest should be tested but also an additional 'placebo notation' (which is, however, not evaluated).

Negative position effects such as getting bored or tired can be avoided if the tasks are as brief and as interesting as possible [17]. Learning is a positive position effect that, however, may lead to skewed results [9]. A *warm-up phase*, i.e., several tasks whose solutions are not evaluated, encapsulates very strong learning in the beginning of an experiment [17]. It also makes participants familiar with the equipment and thus reduces errors in operation.

Measurement errors are not uncommon if dependent variables rely on time. To increase the *reliability* of the gathered data (i.e., to produce the same results from one occasion to another [6]), I have developed the tool `notate`. It is available for downloading from the website http://sourceforge.net/projects/notate. `Notate` fulfils the requirements listed in this section: It shows multiple-choice questions and the descriptions they refer to in random order and records response times, responses, and correctness. `Notate` supports experiments on understandability in general and, since it is an open-source project, can be adapted to specific experimental needs.

3.4 Evaluating an Experiment

The statistical evaluation of experiments on the understandability of notations consists in assessing differences between measures of central tendency (mean, mode, etc.) or dispersion (variance, standard deviation, etc.), proportions, or total numbers. Planning and conducting experiments aim at ascribing as much of these differences as possible to the influence of the independent variable.

Statistical test procedures generally assume that differences in experimental outcomes arise by chance (*null hypothesis*). On the basis of probability distributions, one can calculate the probability of an observed outcome if the null hypothesis proves true. If this probability (*p-value*) is very low (by convention: Smaller than $\alpha = 0.05$ [7]), it is unlikely that an observed difference is accidental. Hence, not the null but the alternative hypothesis is valid, which attributes the difference to some systematic influence of the independent variable(s). Results that require rejecting the null hypothesis ($p < \alpha$) are called *statistically significant*; α describes the probability of making a *Type I* error (rejecting the null hypothesis when it is true [7]).

Numerous statistical test procedures exist; e.g., [16], [17], and [6]. Experimental design and the measurement scale of the dependent variable determine which of them are allowed in evaluating an experiment. Tests that are typically implemented in software packages like SPSS are summarized in Table 2. Non-parametric tests that apply to non-metric variables are usually applicable, since metric variables can be transformed. Such tests remain valid if the assumptions of parametric tests (e.g., certain distributions within the population from which the sample came) are not fulfilled [6].

Concentrating on statistical significance alone can be misleading, because significance is affected by sample size [6]: If the same experiment is conducted independently with varying sample size, the larger sample may yield a statistically significant result, while the smaller one does not. Which result is closer to the empirical 'truth'

depends on the detected effect. *Effect size* expresses the magnitude of a result (the actual difference in the understandability of notations), independently of sample size [6]. Table 3 summarizes the effect size measures that agree to the statistical test procedures of Table 2. It also defines what constitutes a small, medium or large effect. Approximating the effect of non-parametric tests requires samples with $N \geq 25$, where z-scores can be calculated by assuming normal distribution [7].

Table 3. Measuring Effect Size

Effect size measure	**Statistical test procedure**	**Refe-rence**	**Effect**		
			Small	**Medium**	**Large**
$d = \frac{\mu_1 - \mu_2}{\sigma}$ or $r_{es,t} = \sqrt{\frac{t^2}{t^2 + df}}$	t-test*	[7]	0.2	0.5	0.8
$\omega = \sqrt{\frac{\chi^2}{N}}$	χ^2-test	[7]	0.1	0.3	0.5
$\eta^2 = \frac{\sigma_\mu^2}{\sigma^2}$	F-test (ANOVA)	[7], [6]	0.01	0.06	0.14
$r_{es,z} = \frac{\lvert z \rvert}{\sqrt{N}}$	U-test or any other that yields a z-score	[17]	0.1	0.3	0.5

μ_1, μ_2: Group means, σ: Standard deviation. σ^2 (σ_μ^2): Total (Between group means) variance, t, χ^2 (z): Test statistics (z: normal distribution), N: Total number of participants ($N = \Sigma n_i$), *df*: Degrees of freedom.

∗ Between-subjects design: σ of either group, within-subjects design: adjusted σ [6].

None of the investigations of Table 1 reported effect sizes, and mostly the statistics to calculate them ex post (test statistics, mean, standard deviation, or variances) are not given. Without effect sizes, it is not sensible to compare empirical results [6]. [3] and [14] provide at least enough aggregated experimental data to determine the effect sizes, which actually are large ($d = 1.3$ to $d = 2.12$ for the higher correctness of EER [3], and $\eta^2 = 0.15$ for the greater correctness of OOM [14]).

Besides sample size, statistical significance depends on the value α, which was not even stated in [9] and [14]. The larger α is set and the easier significance is achieved. A very small value of α decreases the danger of making a Type I error at the cost of a growing probability β of a *Type II error* (failing to reject the null hypothesis even if it is false) [6]. *Statistical power* is defined as the probability of avoiding a Type II error. As a rule of thumb, it is set to 0.8, corresponding to $\beta = 0.2$ [6]. A small β guarantees that significant differences in the understandability of notations are detected.

Power can be used retrospectively to assess a statistical test or prospectively to determine which sample size is necessary to reveal a large, medium or small effect for a given experimental design and statistical test procedure (e.g., [6], [7]). The sample sizes recommended in Table 2 result from a prospective power analysis.

submitted to: *ER 2007 - 26th Inter. Conference on Conceptual Modeling (ER 2007)*

4 Applying the Guideline to Test SAP's Notations

Whether the graphical or the textual notation of Fig. 1 is easier to understand is an analytical research question. The contradictory results of Table 1 call for a two-tailed hypothesis; whether a one-tailed hypothesis can be formulated depends the particular data gathered in the experiment.

Notation constitutes the independent variable, with its levels comprising of a) and b) of Fig.1 and a placebo (UML class diagrams). Since the tool `notate` was applied in the experiment, response time and the number of correct answers form the dependent variables.

A within subjects-design can very simply answer my analytical research question. 15 participants are necessary to detect a large effect ($d = 0.8$) by the corresponding paired t-test (two-tailed) with a power of 0.8 and $\alpha = 0.05$ [6]. In contrast, under the same conditions (d, α, β), non-parametric tests need 16 to 24 subjects [6]. $N = 30$ - the typical sample size used in Table 1 - additionally buffers against outliers.

For a small financial incentive, 42 subjects (35 male, 7 female) volunteered to participate in the experiment. Six of them already had graduate degrees in computer science, the other 36 were students (undergraduates: 17, graduates: 19) of computer science, computational visualistics, data and knowledge engineering, and business informatics from three universities. The sample size relevant to the statistical test procedures is smaller than 42, as I had to remove outliers (detected by SPSS® boxplots).

The experiments were conducted in groups over several days at the same time; the instructor did not change. Each participant was seated at a PC where the tool `notate` was installed. The participants received written instructions and explanations of the notations to be tested (levels of the IV). After 15 minutes, the warm-up phase began. When all participants had finished this phase, the actual experiment started. The subjects were told to answer all questions correctly and as quickly as possible. During both the warm-up and the experiment, the questions appeared randomly on each PC.

The warm-up phase consisted of six questions: Two for each level of the IV, respectively related to a small and a large description. The experimental data was gathered from another 27 questions. All questions referred to diagrams and tables that describe SAP's latest application system. The size of the diagrams and tables in the experiment (Table 4) agrees with the complexity of SAP's real-world descriptions.

The questions were in multiple-choice form, dealt with cardinalities and had only one correct solution[6]. To check whether or not the questions were clearly formulated, I conducted a pre-test involving five participants; this data is omitted in Table 4.

Table 4 shows the descriptive statistics. Concerning the correctness of answers, there is no significant difference between the notations a) and b) of Fig. 1 ($t = -1.481$, $df = 39$, $N = 40$, $p = 0.147$, two-tailed). The non-parametric sign test leads to the same conclusion ($N = 40$, $p = 0.166$; one-tailed in favour of the textual notation), whereas the Wilcoxon matched-pairs signed rank test shows a significant superiority of the textual notation at $\alpha = 0.1$ ($z = -1.470$, $N = 40$, $p = 0.085$, one-tailed).

[6] Referring to Fig. 1b): *What does the description express?* a) In each expense report, at least one stay location is given; b) Expense reports, where the name of the stay location is missing, are allowed; or c) Each expense report must contain exactly one stay location. Answer b) is correct.

On the other hand, a significantly shorter response time indicates that the textual notation is easier to understand (N = 39): The paired t-test detects a significant difference ($t = 5.625$, $df = 38$, $p = 0.00$, two-tailed), while both one-tailed tests – the sign test ($z = -4.163$, $p = 0.00$) and the Wilcoxon test ($z = -4.585$, $p = 0.00$) - favour the textual notation. All statistics are calculated by the program SPSS® version 14.0.1.

The t-test could not detect a significant difference in correctness because the effect is small ($d = 0.30$; $\sigma_{adjust} = 0.56$); for the Wilcoxon-test, it is almost medium ($r_{es,z} = 0.23$). The significantly shorter response time for the textual notation should also not be overemphasized, since it results from only a medium effect in the t-test ($d = 0.60$; $\sigma_{adjust} = 5.05$); in the Wilcoxon-test, however, the effect is large ($r_{es,z} = 0.73$.).

The notations SAP AG has invented are equally capable of unambiguously describing the application system. Unambiguity is one of the most critical aspects in the model-driven development process. The textual notation is just easier to read.

Table 4. Summary of Input and Output of the Experiment

Notation level	**Graphical** (Fig. 1a)	**Textual** (Fig. 1b)	**Placebo**
Size of the diagrams underlying the questions			
Very small (ca. 4 attributes/classes)	2	2	2
Small (ca. 9 attributes/classes)	2	2	2
Medium (ca. 14 attributes/classes)	4	4	2
Large (> 18 attributes/classes)	2	2	1
Sum	10	10	7
Descriptive statistics			
Number of correct answers	$\mu = 9.63$, $\sigma = 0.59$	$\mu = 9.80$, $\sigma = 0.52$	-
Response time (seconds)	$\mu = 26.41$, $\sigma = 5.12$	$\mu = 23.37$, $\sigma = 4.98$	-

5 Contribution of this Paper

I could have concluded this paper by emphasizing the precision of graphical notations and their bad understandability (compared to textual ones). In the light of all the papers that either glorify the "easy to understand" graphical notations (e.g., [13] [19]) or complain about their vagueness (e.g., [20]), these results would be quite surprising. But they miss my contribution:

This paper presents a guideline for conducting experiments on understandability – not only of notations, but also of the constructs of higher-level programming languages, etc. This guideline is not provided as another bundle of paper, but supported by and implemented in the tool `notate`, which can be modified to fit specific experimental situations. `Notate` can be applied quite broadly, helps researchers to avoid common mistakes in empirical research (e.g., neglecting carry-over or warm-up effects) and makes the conducted experiments comparable.

Comparability requires standardizing the statistical assumptions ($\alpha = 0.05$ and $\beta = 0.2$) and the reporting of statistics: Following APA, mean, standard deviation or variance, test statistic, sample size, degrees of freedom and p-value should always be given (http://apastyle.apa.org/). Standardization enables *meta-analysis* that pools the

submitted to: *ER 2007 - 26th Inter. Conference on Conceptual Modeling (ER 2007)*

results of prior studies to create an integrated view on the empirical situation [17]. Perhaps meta-analysis reveals the actual superiority of some notational style.

In acknowledging my results, SAP AG admitted that the graphical notation became necessary, because the tool containing the repository of the new application system is unable to display tables. So, redundant modelling is unavoidable here. SAP AG will apply my guideline in an upcoming experiment to test the understandability of alternative notations for business processes before they are released to public.

Acknowledgments I would like to thank SAP AG, especially Peter Zencke and Stefan Kaetker, for allowing this investigation and the use of real-world examples. Thanks also to Baoguo Chen from Beijing Normal University for discussions about the nature of psychological experiments and for the inspiration to create notate.

References

1. Anderson, J.R.: Cognitive Psychology and its Implications. 5th ed., Worth, New York: (2000)
2. Batra, D., Hoffer, J.A., Bostrom, R.P.: Comparing Representations with Relational and EER Models. Comm. of the ACM 33 (1990) 2, 126–139
3. Bock, D., Ryan, T.: Accuracy in Modeling with Extended Entity Relationship and Object Oriented Data Models. J. of Database Management 4 (1993) 4, 30-39
4. Bourne, L.E., Battig, W.F.: Complex Processes, in Sidowski, J.E. (ed.): Experimental methods ans instrumentation in psychology. McGraw-Hill, New York et al. (1966), 541-576
5. Chan, H.C.: Naturalness of Graphical Queries Based on the Entity Relationship Model. J. of Database Management 6 (1995) 3, 3–13
6. Clark-Carter, D.: Quantitative psychological research. 2nd ed., Psychology Press, Hove (2004)
7. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences. 2nd ed., Erlbaum, Hillsdale (1988)
8. Hayes, P.J.: Some Problems and Non-Problems in Representation Theory. Proc. of the AISB Summer Conference. University of Sussex (1974) 63-79
9. Jamison, W., Teng, J.T.C.: Effects of Graphical Versus Textual Representation of Database Structure on Query Performance. J. of Database Management 4 (1993) 1, 16–23
10. Lee, H., Choi, B.G.: A Comparative Study of Conceptual Data Modeling Techniques. J. of Database Management 9 (1998) 2, 26-35
11. Mook, D.: Classic experiments in psychology. Greenwood, Westport (2004)
12. Moore, R.C.: The Role of Logic in Knowledge Representation and Commonsense Reasoning. Proc. AAAI '82. AAAI, Menlo Parc (1982) 428–433
13. Larkin, J. H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11 (1987) 65–99
14. Palvia, P.C., Liao, C., To, P.-L.: The Impact of Conceptual Data Models on End-User Performance. J. of Database Management 3 (1992) 4, 4-15
15. Patig, S.: Evolution of Entity-Relationship Modelling. Data & Knowledge Engineering 56 (2006), 122-138
16. Robinson, P.W.: Fundamentals of Experimental Psychology, 2nd ed., Prentice-Hall, Englewood Cliffs (1981)
17. Sarafino, E.P.: Research Methods: Using Processes and Procedures of Science to Underssand Behavior. Pearson, Upper Saddle River (2005)
18. Shimojima, A.: The Graphic-Linguistic Distinction: Exploring Alternatives. Artificial Intelligence Review 15 (2001) 5–27
19. Sloman, A.: Interactions Between Philosophy and Artificial Intelligence. Artificial Intelligence 2 (1971) 209–225
20. ter Hofstede, A.H.M.; van der Weide, T.P.: Formalization of techniques: chopping down the methodology jungle. Information and Software Technology 34 (1992) 1, 57-65