Empirical Software Engineering, 10, 367–394, 2005.

Methodology Support in CASE Tools and Its Impact on Individual Acceptance and Use: A Controlled Experiment*

JÖRG ZETTEL j.zettel@computer.org
Hessische Zentrale für Datenverarbeitung (HZD), Postfach 3164, 65021 Wiesbaden, Germany

Editor: Stan Rifkin

Abstract. This paper reports the results of a controlled experiment undertaken to investigate whether the methodology support offered by a CASE tool does have an impact on the tool's acceptance and actual use by individuals. Subjects used the process modelling tool SPEARMINT to complete a partial process model and remove all inconsistencies. Half the subjects used a variant of SPEARMINT that corrected consistency violations automatically and silently, whilst the other half used a variant of SPEARMINT that told them about inconsistencies both immediately and persistently but without automatic correction. Measurement of acceptance and prediction of actual use was based on the technology acceptance model, supplemented by beliefs about consistency rules. The impact of *form of automated consistency assurance* applied for hierarchical consistency rules was found to be significant at the 0.05 level with a type I error of 0.027, explaining 71.6% of the variance in CASE tool acceptance. However, intention to use and thus predicted use was of the same size for both variants of SPEARMINT, whereas perceived usefulness and perceived ease of use were affected contrarily. Internal validity of the findings was threatened by validity and reliability issues related to beliefs about consistency rules. Here, further research is needed to develop valid constructs and reliable scales. Following the experiment, a small survey among experienced users of SPEARMINT found that different forms of automated consistency assurance were preferred depending on individual, consistency rule, and task characteristics. Based on these findings, it is recommended that vendors should provide CASE tools with *adaptable* methodology support, which allow their users to fit automated consistency assurance to the task at hand.

Keywords: Computer-aided software engineering, methodology support, consistency assurance, technology acceptance, process modelling.

1. Introduction

1.1. Motivation

Computer-aided software engineering (CASE) is a collective term commonly used to denote software tools that are used throughout the software life-cycle processes (ISO 12207, 1995). The spectrum of possible CASE tools is broad. It ranges from simple text editing tools to complex project support environments (Fuggetta, 1993).

Often, CASE tools provide functionality to support their users in applying specific software engineering methods. To support for example the application of a structured

*This work originates from the author's time at the Fraunhofer Institute for Experimental Software Engineering (IESE), Sauerwiesen 6, 67661 Kaiserslautern, Germany.

analysis and design method, a CASE tool must offer the possibility to create and edit data flow diagrams. This functionality has been generally named *methodology support* (Vessey et al., 1992; Jankowski, 1997).[1]

Vendors and proponents of CASE tools with methodology support commonly claim these tools would increase the productivity of their users and improve the quality of the resulting products. However, published empirical evidence on the *actual* impact of CASE tools is rare and controversial (Henderson and Cooprider, 1990; Iivari, 1996; Glass, 1999).

On one hand, CASE tools are often introduced in combination with a new method. Thus, observations exhibit the *combined* effect of at least the method and the CASE tool. On the other hand, few studies take actual *use* of the CASE tools adequately into account (Henderson and Cooprider, 1990; Iivari, 1996). Thus, unreported variations in actual use could be an explanation for diversity in findings as well (DeLone and McLean, 1992; Goodhue, 1995).

Another look at the literature confirms that actual use is indeed a problem with today's CASE tools. For example, Fichman and Kemerer reported of a study showing that about more than 50% of the acquired CASE tools will probably never be used (Fichman and Kemerer, 1999).

Several technical and social factors come into question as reasons for the humble utilisation of CASE tools. Insufficient *task-technology fit* is one of them (Goodhue and Thompson, 1995). Especially the methodology support offered by today's CASE tools has been proposed as a technology characteristic that does not fit the typical characteristics of tasks in a modelling situation, where creativity, understanding, intuition etc. play a major role (Jarzabek and Huang, 1998). However, there is only small empirical evidence that this proposition applies (Chau, 1996).

1.2. Research Goal

The goal of this research was to provide sound empirical evidence in support of the hypothesis that the methodology support offered by a CASE tool does have an impact on the individual acceptance and use of the tool. Therefore, a controlled experiment was performed. Its results confirmed the hypothesis.

In the context of the research, only students were available as subjects. Missing CASE tool experience, a small sample size, and a short time-frame had to be accepted up front, running the risk of threatening external and internal validity of the findings. But because of the lack of empirical research in the area, the experiment was considered to be an interesting investigation nonetheless.

1.3. SPEARMINT

The experiment was performed using a single CASE tool with configurable methodology support, instead of comparing different CASE tools with each other. Thus, uncontrollable effects due to differences in human–computer interfaces were avoided (Brooks et al., 2000a).

For this research, SPEARMINT 5.1 as of May 2001 was used.[2] SPEARMINT is a CASE tool for descriptively modelling software development processes (Becker-Kornstaedt et al., 1999). It comes with a large set of methodological consistency rules (Zettel, 2001), and provides configurable functionality for automated consistency checking.

SPEARMINT masters nine different *forms of automated consistency assurance* (Table 1). For each individual consistency rule, the user can configure the form of automated consistency assurance to by applied by SPEARMINT. The effect of these forms on the tool's behaviour is as follows:

- With form IGN, a consistency rule is effectively disabled since violations are completely ignored.

- With forms CP and CT1–CT4, a consistency rule may be violated, which results in inconsistencies being tolerated in the model. These inconsistencies are either detected automatically as soon as they are created (CT1–CT4), or they are detected on explicit request of the user only (CP). In case of automatic detection, the form also determines how the user is informed about inconsistencies. On one hand, he or she may be interrupted by a information dialog (CT3, CT4). On the other hand, inconsistencies may be visualised by appropriate graphical means in the model's diagrammatic display (CT2, CT4).

- With forms CC1 and CC2, violations of consistency rules are automatically detected and immediately corrected. With CC2, the user is informed by a dialog prior to automatic correction.

- With form PRE, the user is prevented to violate a consistency rule. This is done by temporarily disabling those user-level functions, whose execution in the current state of the model would result in an inconsistency.

Table 1. Forms of automated consistency assurance.

			Intervention form	
Form	Enforcement semantics	Intervention strategy	Interrupt	Display
IGN	ignore	–	–	–
CP	tolerate	passive	no	yes
CT1	tolerate	active	no	no
CT2	tolerate	active	no	yes
CT3	tolerate	active	yes	no
CT4	tolerate	active	yes	yes
CC1	correct	active	no	–
CC2	correct	active	yes	–
PRE	prevent	–	–	–

A more detailed discussion of forms of automated consistency assurance and a description of how to use them for the specification and construction of adaptable consistency assurance in CASE tools can be found in (Zettel, 2003).[3]

2. Theory

Individual acceptance and use of CASE tools is a special case of individual acceptance and use of computer technology in general. For this, a solid theory with reliable measurement scales has been developed and empirically validated by management information systems (MIS) research over the past 15–20 years. This theory is known as the *technology acceptance model*.

Methodology support in CASE tools has been examined by only a few studies. These were focused on the impact of methodology support on productivity and quality. They didn't examine acceptance and use of CASE tools, even though utilisation is a prerequisite to performance impacts (DeLone and McLean, 1992; Goodhue and Thompson, 1995).

The following section introduces the technology acceptance model and describes prior studies on methodology support. This is followed by a description of the research model and the research hypotheses underlying this work.

2.1. Prior Work

2.1.1. Technology Acceptance Model (TAM)

The technology acceptance model has been originally proposed by Davis more than 15 years ago (Davis, 1986, 1989; Davis et al., 1989). Its goal is "to provide an explanation of the determinants of computer acceptance that is general, capable of explaining user behavior across a broad range of end-user computing technologies and user populations, while at the same time being parsimonious and theoretically justified" (Davis et al., 1989; p. 985).

The theoretical foundation of TAM is the *theory of reasoned action*, a well-known model from social psychology (Fishbein and Ajzen, 1975). According to the theory of reasoned action, a person's behaviour is determined by his or her intention to perform the behaviour, which itself is influenced by the person's attitude and subjective norm concerning that behaviour (Figure 1). Attitude is determined by a person's beliefs about the expected consequences of the behaviour, whereas subjective norm is determined by a person's normative beliefs about the behaviour, i.e., his or her perceived expectations of referent individuals or groups and the motivation to comply with them.

The technology acceptance model is a specialisation of the theory of reasoned action, applied to individual acceptance and use of computer technology (Figure 2). It has been designed in such a way that it can be used both to investigate properties of a computer system and to predict its actual use by individuals. At its core are two selected individual beliefs: *perceived usefulness* and *perceived ease of use*. In combination, they can be used

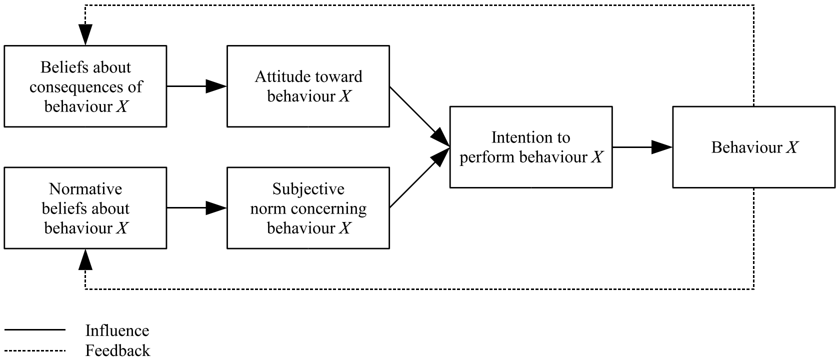


Figure 1. Theory of reasoned action.

to explain and predict the probability of an individual actually using a software system for a given task:

> Perceived usefulness is defined as the prospective user's subjective probability that using a specific application system will increase his or her job performance within an organisational context. Perceived ease of use refers to the degree to which the prospective user expects the target system to be free of effort. (Davis et al., 1989; p. 985)

TAM provides multi-item scales for measuring perceived usefulness and perceived ease of use, which are easily adaptable to the software system and use context at hand, while at the same time being valid and reliable (Davis and Venkatesh, 1996; Davis et al., 1989). Thus, scales need not be developed and validated again for different research situations, which is a great practical advantage of TAM.

TAM and its scales have been applied and empirically validated repeatedly (Davis, 1993; Adams et al., 1992; Segars and Grover, 1993; Szajna, 1994; Morris and Dillon, 1997; Henderson et al., 1998; Laitenberger and Dreyer, 1998; Roberts and Henderson, 2000). Several comparisons with other approaches have confirmed TAM's validity as well (Davis et al., 1989; Mathieson, 1991; Taylor and Todd, 1995).[4]

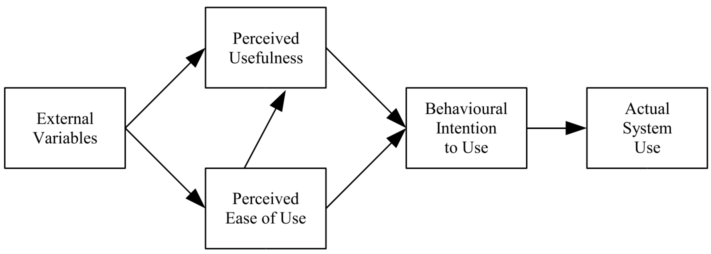


Figure 2. Technology acceptance model.

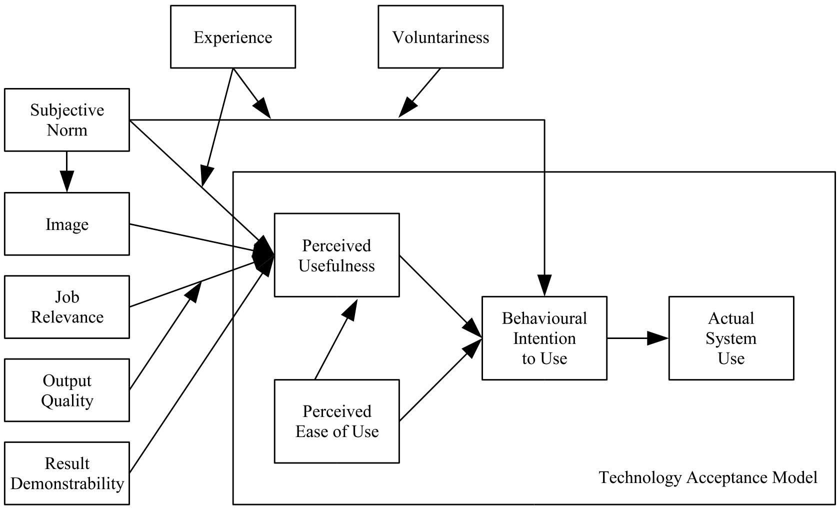


Figure 3. Determinants of perceived usefulness.

Over time, several factors have been identified that have an impact on perceived usefulness (Venkatesh and Davis, 2000; Figure 3) and perceived ease of use (Venkatesh and Davis, 1996; Figure 4). Most of these factors come as individual beliefs, and mediate external influences on technology acceptance. Two notable exceptions to this are *objective usability* and *computer self-efficacy*. The first relates to a software system's usability measured in terms of the keystroke-level model (Card et al., 1987). The second relates to a person's self-efficacy belief regarding computers in general, which is independent of the software system under investigation.

For researchers, knowledge of the determinants of perceived usefulness and perceived ease of use is essential when designing an experimental study about individual acceptance and use of software systems. Depending on the situation at hand, a factor could pose a threat to validity and should be controlled adequately, or he could mediate the

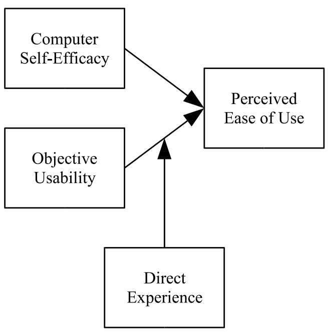


Figure 4. Determinants of perceived ease of use.

effect of the experimental variable(s) and should be included in the statistical analysis. The researcher needs to consider this question as early as possible.

2.1.2. Methodology Support

The starting point for empirical research on methodology support in CASE tools was a survey reported by Vessey, Jarvenpaa and Tractinsky (Vessey et al., 1992). The survey was performed in summer 1989 and resulted in a characterisation of twelve commercially available CASE tools with regard to their support for structured analysis and design methods.

Some years later, two of these CASE tools—a very flexible one and a very restrictive one—were used in an empirical study reported by Jankowski (Jankowski, 1997). The goal of the study was to investigate the relationship between methodology support and product quality. The findings were unconvincing, but there was some evidence that internal consistency rules are adhered to regardless of the type of methodology support, whereas hierarchical consistency rules are adhered to more likely in the presence of restrictive methodology support.

Brooks, Campbell and Scott explained the lack of findings reported by Jankowski with differences in the human–computer interface of the two CASE tools compared (Brooks et al., 2000a). To increase the validity of future studies, they recommend using a single CASE tool with configurable methodology support instead of comparing different CASE tools.

From 1996 to 1999, the international research project CADPRO tried to empirically explore the relationship between methodological constraints that are present in a CASE tool and the use of the tool (Jeffery and Offen, 1999). The target was to provide advice on how to design such constraints in commercial CASE tools.

CADPRO's starting point was a survey reported by Day (Day, 1996), which came to the conclusion that the more control a CASE tool exercises according to its users' perceptions, the more they are unsatisfied with the tool. Based on this conclusion, Day, Ahuja and Scott developed a theoretical model, which became the research model of CADPRO (Day et al., 1997). This model links individual differences, task characteristics, and constraint characteristics (inputs) with beliefs about constraints and attitudes toward them. These two in turn influence user productivity and product quality (outputs) via multiple steps (Figure 5). The CADPRO research model had been influenced by TAM and was based on the assumption that no direct relationships between input and output constructs exist; psychological constructs are assumed to mediate all effects instead.

Several studies were performed to empirically validate the CADPRO research model:

– Brooks and Scott reported of two prestudies (Brooks and Scott, 2001), which confirmed that individual beliefs on (the presence of) constraints varied significantly, and that users' mental models of CASE tools acted as perceptual filters for these beliefs.

– Brooks, Takada and Scott reported of two pilot studies in preparation for a planned experiment (Brooks et al., 1999), which revealed that traditional metrics for

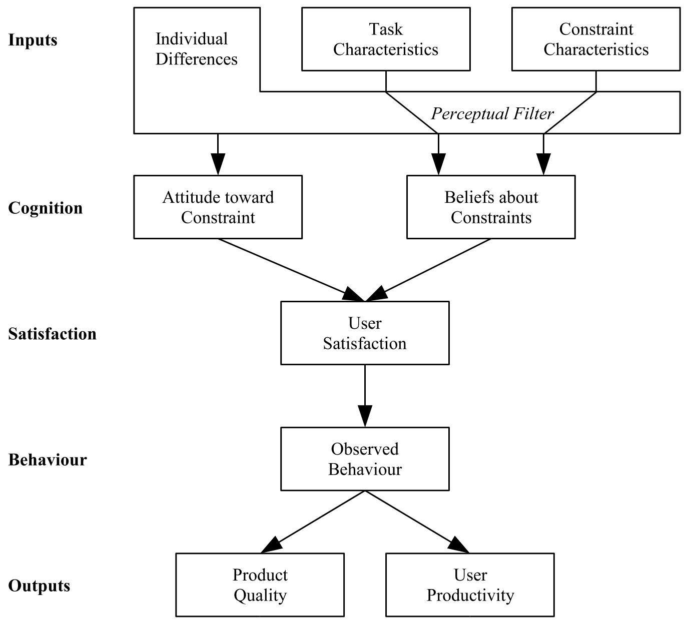


Figure 5. CADPRO research model.

measuring productivity and quality were not applicable to the kind of small analysis models typically produced in experiments. It was suggested to use performance profiles, i.e., delivery and constraint profiles, instead, which "chart with time how the work product grows and how instances of the methodological constraint violations come and go and persist to the end of a work session" (Brooks et al., 1999, p. 270).

– Brooks et al. reported of an experiment, in which only one half of the subjects were pointed at violations of internal consistency rules when using a CASE tool (Brooks et al., 2000b). Instead of testing a formal null hypothesis, automatically recorded delivery and constraint profiles were analysed. Productivity and quality varied strongly, but did not follow a clear pattern. An analysis of the delivery profiles indicated productivity losses that were caused by human–computer interface constraints. In total, some evidence was found for the validity of particular aspects of the CADPRO research model. At the same time, it became clear that the influence of human–computer interface constraints and of individual working speed must be taken into account.

Empirical findings on methodology support in CASE tools can be found in other research areas as well, namely in the area of automated critiquing (Fischer et al., 1991; Robbins and Redmiles, 1999):

– Silverman and Mehzer reported experimental evidence that automated critics helped programmers remove faults (Silverman and Mehzer, 1997). Full critiquing with indication of faults, explanation of errors, and suggestion of corrections sped up the activity by a factor of 3.5 compared to mere debugging, i.e., indication of faults only. 53% of the users found the automated critic useful and liked to use it; 16% found it useful, but didn't trust it; 31% didn't like to use it because it was time consuming and slowed them down.

2.2. Research Model

The research model for this work is assembled of three components: form of automated consistency assurance, CASE tool acceptance, and beliefs about consistency rules (Figure 6).

- *Form of automated consistency assurance* is a variable that is used to characterise the methodology support provided by a CASE tool (Table 1).[5]

- *CASE tool acceptance* is a subset of TAM, which is used to predict actual use of the tool. The subset contains the core TAM variables perceived usefulness, perceived ease of use, and intention to use. These are complemented by those influencing factors that needed to be controlled for in the context of this research: job relevance, output quality, result demonstrability, and computer self-efficacy.

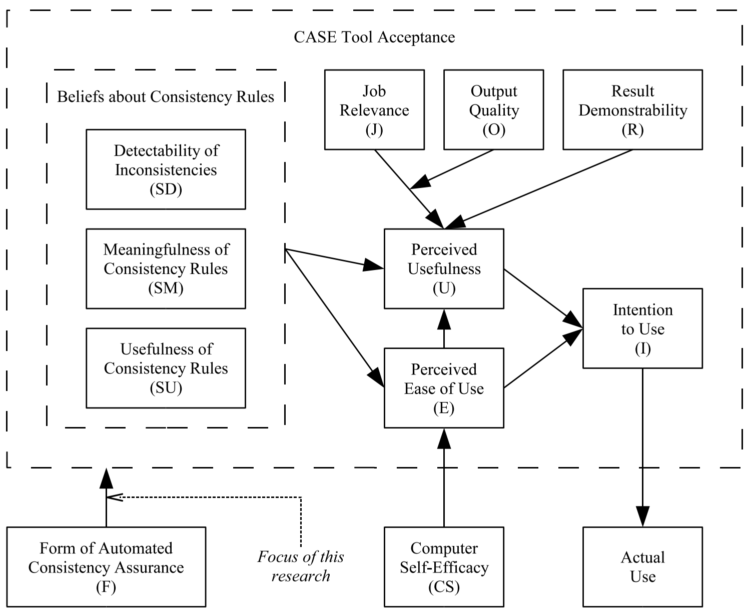


Figure 6. Research model.

– *Beliefs about consistency rules* is a set that contains the three variables detectability of inconsistencies, meaningfulness of consistency rules, and usefulness of consistency rules. These variables were added as additional influencing factors. The rationale was that results from the CADPRO research suggested that the effect of methodology support on CASE tool acceptance is possibly mediated by such beliefs.

The focus of this research was on investigating the relationship between form of automated consistency assurance and CASE tool acceptance.

2.3. Research Hypotheses

For the experiment, three out of SPEARMINT's 48 consistency rules were chosen. These rules were related to hierarchical consistency of activities being refined by product flow graphs. Based on prior research it was expected that assuring hierarchical consistency automatically or not would have a significant effect on CASE tool acceptance (Brooks and Sobczak, 1999).

Three variants of SPEARMINT were prepared, which differed only with respect to the form of automated consistency assurance that was applied to the three chosen rules:

– Variant A used form CP (tolerate passively; display).

– Variant B used form CC1 (correct actively; don't interrupt).

– Variant C used form CT4 (tolerate actively; interrupt and display).

The aim of the controlled experiment was to compare variants B and C of SPEARMINT with respect to CASE tool acceptance and predicted use. Variant A was used for training purposes only.

Since methodology support was considered to impact CASE tool acceptance, the primary research hypothesis was as follows:

Hypothesis 1. CASE tool acceptance of variant B and variant C of SPEARMINT differs.

Since form CC1 was considered to be more comfortable than form CT4, the secondary hypothesis was as follows:

Hypothesis 2. Predicted use is higher for variant B than for variant C of SPEARMINT.

In the experiment, external effects were empirically and statistically controlled for to the greatest possible extent. Therefore, if the primary hypothesis was supported, the difference in CASE tool acceptance could most likely be attributed to the form of automated consistency assurance applied for hierarchical consistency. This would provide sound empirical evidence for the existance of a relationship between methodology support and CASE tool acceptance. If the secondary hypothesis was supported as well, it would provide a starting point for quantifying this relationship.

The following sections describe the controlled experiment.

3. Method

3.1. Sample

The experiment was performed with students attending a software engineering class at the University of Kaiserslautern, Germany, in the summer term of 2001. Out of about 30 to 40 students, who attended the lectures regularly, 28 students voluntarily participated in the experiment. Most of the participants were in their third, forth, or fifth year of studying computer science.

3.2. Experimental Design

The experiment was to be performed within a single session of 90 minutes. There was no possibility of prior training. Therefore, the experiment itself was set up as a training in the use of SPEARMINT. The learning effects to be expected were explicitly taken into account by the experimental design and the statistical analysis.

Another expected effect could be dealt with only optimistically: Venkatesh and Davis have shown that perceived ease of use always depends on an individual's computer self-efficacy and is influenced by the tool's objective usability only after the subject has used the tool for at least some time (Venkatesh and Davis, 1996). Whether the available 90 minutes would suffice to observe measurable changes due to tool characteristics was unclear. But since there was no way to extend the time-frame, this risk had to be taken.

For the controlled experiment, a covariance design with repeated measures was chosen, namely a pretest-posttest two-groups completely randomised experimental design (Table 2). This design permitted statistical control both of pretest variability among subjects—regarding computer self-efficacy in particular—and of learning effects per subject.

First, subjects were randomly assigned to one of the two groups. Both groups started by answering questionnaire O_1 about sample characteristics, and worked through exercise X_1 to familiarise themselves with the fundamental concepts and functions as well as the human–computer interface of the CASE tool.

Then, both groups worked through the "real" exercise X_2 with variant A of the tool (X_{2A}). The task was to complete a given process model and remove all inconsistencies manually (due to form CP being used in variant A). This was followed by answering questionnaire O_2 to gather acceptance data the first time (pretest).

Table 2. Experimental design.

R	O_1	X_1	X_{2A}	O_2	X_{2B}	O_2
R	O_1	X_1	X_{2A}	O_2	X_{2C}	O_2

Finally, both groups worked through exercise X_2 again, but this time with active support by the tool. One group used variant B of the tool (X_{2B}), which corrected all inconsistencies automatically and silently (form CC1). The other group used variant C of the tool (X_{2C}), which interrupted the user upon creation of an inconsistency but didn't fix it automatically (form CT4). This was followed by answering questionnaire O_2 a second time (posttest).

Whether a subject used variant B or C of SPEARMINT was captured by the experimental variable F (corresponding to the form of automated consistency assurance).

As Figure 6 points out, CASE tool acceptance was measured by the dependent variables I (intention to use), U (perceived usefulness), E (ease of use), J (job relevance), O (output quality), and R (result demonstrability), complemented by SD (detectability of inconsistencies), SM (meaningfulness of consistency rules) and SU (usefulness of consistency rules), i.e., the beliefs about consistency rules. Pretest and posttest measurements of these dependent variables were coded by a time variable T.

In addition, the independent variables CS (computer self-efficacy) and ES (prior experience with SPEARMINT: yes or no) were measured to capture the variability among subjects prior to the experiment.

Most variables were measured with one or more items based on a seven-point agreement scale. For data analysis, total disagreement was coded as 1 and total agreement as 7. Thus, 4 marks the scale's neutral point.

Detectability of inconsistencies (SD) is a special case. The SD item asks whether inconsistencies in the model can be found without the CASE tool. Agreement represents a *negative* perception of automated consistency assurance in this case. Therefore, the inverse variable SD' (SD' = 8 − SD) was used in most of the statistical analyses instead of SD.

3.3. Experimental Material

In one of the lectures preceding the experiment, participants were given a short introduction into process modelling as a normal part of the course. Thus, it was assumed that they were neither trained in process modelling nor experienced in using SPEARMINT. Working through exercise X_1 was intended to enable all participant in equal measure to use the tool.

Handouts consisted of a sheet of instructions regarding the course of the experiment, three exercises (one time X_1 and two times X_2), and three questionnaires (one time O_1 and two times O_2). Since these were in English, a German translation of several English words was given as well. The documents had been pretested with one student, resulting in small improvements.

The CASE tool and all necessary data had been installed on the available computers prior to the experiment. On neighbouring computers, different variants of SPEARMINT (i.e., B or C) were installed. 30 computers in two rooms were available. Each computer was given a number, which was used both for random assignment of subjects to computers (and groups) and anonymous identification of subjects on questionnaires.

3.4. Experimental Procedure

Working through the exercises and filling in the questionnaires was done by each subject on his own. Teamworking was not permitted. Completed questionnaires were to be delivered immediately to the attending supervisors. In addition, completed but not yet delivered questionnaires were collected in regular intervals.

On delivery of a questionnaire to a supervisor, the presence of the subject identification number was immediately checked. Missing answers on other questions were accepted.

Two subjects didn't manage to complete the exercises within the available time and were excluded from the analysis. Another subject experienced a fatal crash of the tool and a total loss of data just before finishing the last exercise. This subject's frustration became so obvious on the final questionnaire, that it was excluded from the analysis as well.

3.5. Data Analysis

Matching the experimental design, a multivariate analysis of covariance (MANCOVA) with repeated measures was performed for testing the hypotheses (Table 3). With this statistical technique, both learning effects in the dependent variables (pretest–posttest differences) and variability in the control variable CS could be filtered out to unsheathe the effect of the experimental variable F.

All statistical analyses were performed with STATISTICA '99 Edition.

4. Results

In the following Section 4.1, the data set is described Psychometric properties of multi-item scales and descriptive statistics for all variables are presented. In addition, variables are tested for normal distribution and correlations. Following that, data set reduction is discussed in Section 4.2. Statistical hypotheses are stated and tested in Section 4.3. Two alternative MANCOVAs were performed, resulting in different findings.

4.1. Data Set

4.1.1. Psychometric Properties

For all multi-item scales with at least three items (i.e., CS, U, E, and R) Cronbach's coefficient alpha (α) was calculated to determine their internal-consistency reliability

Table 3. MANCOVA—design.

DESIGN:	2-way MANCOVA, fixed effects
DEPENDENT:	9 variables (repeated measures): I, U, E, J, O, R, SD', SM, SU
COVARIATE:	1 variable: CS
BETWEEN:	F (2 values): variant B, variant C
WITHIN:	T (2 values): pretest, posttest

Table 4. Cronbach's α for multi-item scales.

Variable	Cronbach's α
CS	0.877
U	0.748
E	0.782
R	0.850

(Table 4). According to (Morris and Dillon, 1997), researchers prefer α values of 0.6 or greater for behavioural studies. In this case, all α values were at least 0.748, i.e., the multi-item scales used were sufficiently reliable. The α values found here were comparable to those found in other TAM studies.

4.1.2. Descriptive Statistics

28 subjects took part in the experiment, three of which were excluded from the analysis due to invalid data (Section 3.4). Out of the remaining 25 subjects, 14 belonged to the first group (F = B), and 11 belonged to the second group (F = C). Two subjects in the first group did have prior experience with SPEARMINT. In both groups, most subjects reported positive computer self-efficacy beliefs (Figure 7).

For working through the three exercises, the subjects needed about 13, 25, and 12 minutes, respectively (Figure 8). Working through exercise X_2 the second time happened with increased performance and reduced variance in both groups. All delivered solutions of exercise X_{2B} were consistent, whereas two subjects delivered solutions of exercise X_{2C} that still contained inconsistencies.

In both groups, acceptance of SPEARMINT was rather positive before the treatment (Figure 9) as well as after the treatment (Figure 10). The same applied to the beliefs about consistency rules and their automated support. In both groups, a slight increase in CASE tool acceptance was observed from pretest to posttest measurement.

4.1.3. Distributions

To test the variables for normality, a Shapiro–Wilks' W test was performed. Results indicated that only about half of the variables were normally distributed.

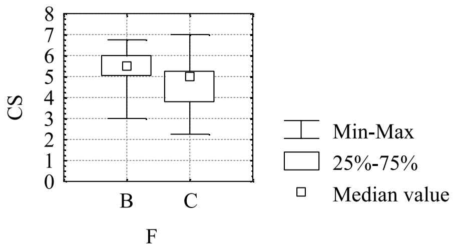
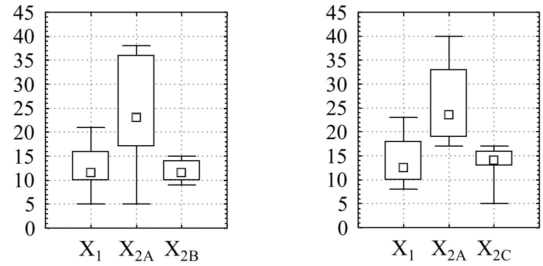


Figure 7. Computer self-efficacy (CS).



a. First group (F = B). b. Second group (F = C).

Figure 8. Time needed for working through the exercises.

4.1.4. Correlations

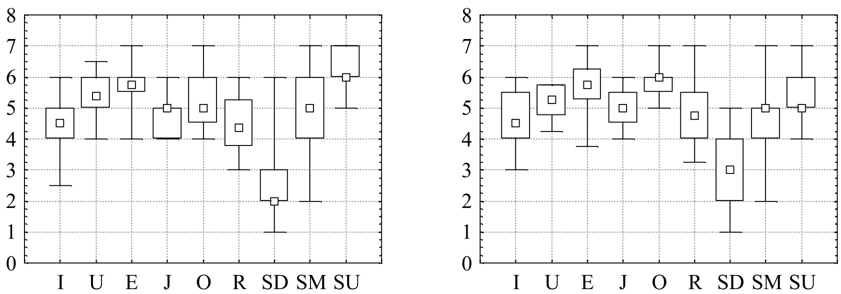
Spearman R, a non-parametric correlation coefficient very similar to the standard Pearson product moment correlation coefficient, was computed to explore relationships between the variables.

To compute Spearman R for correlations among CASE tool acceptance variables, pretest and posttest measurements were pooled (Table 5). The results confirm the research model with three exceptions:

1. TAM postulates a significant correlation between result demonstrability and perceived usefulness, but there isn't. Instead, result demonstrability is significantly correlated to job relevance and output quality, and particularly to intention to use directly.

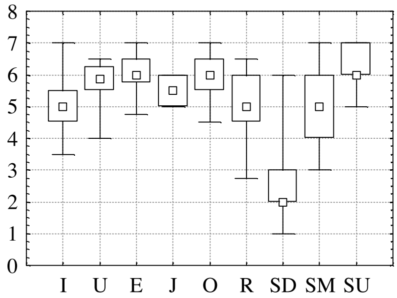
2. Job relevance and output quality are significantly correlated to perceived ease of use, even though TAM does not suggest so.

3. Beliefs about consistency rules are not correlated to perceived usefulness and perceived ease of use directly. Instead, detectability of inconsistencies and usefulness of

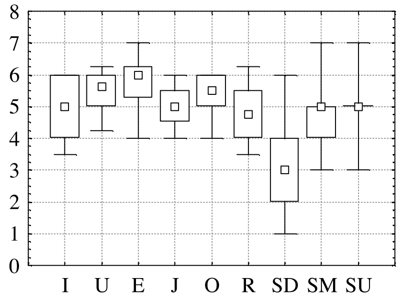


a. First group (F = B). b. Second group (F = C).

Figure 9. CASE tool acceptance before the treatment (pretest).



a. First group (F = B).



b. Second group (F = C).

Figure 10. CASE tool acceptance after the treatment (posttest).

consistency rules are significantly correlated to meaningfulness of consistency rules only, which in turn is particularly correlated to result demonstrability.

Spearman R was also computed for correlations of computer self-efficacy with CASE tool acceptance variables during pretest and posttest (Table 6). The results confirm the research model especially in that the influence of computer self-efficacy on perceived ease of use is high at pretest—explaining about 70 % of the variance—and decreases during the treatment.

4.2. Data Set Reduction

An outlier analysis was performed, but no outliers could be identified.

The two subjects with prior experience in SPEARMINT were not considered outliers since their data fit very well to those of the other subjects. Nonetheless, all statistical analyses reported in the following section were repeated excluding those two subjects, but the results remained the same.

Table 5. Spearman R for CASE tool acceptance.

	U	E	J	O	R	SD	SM	SU
I	0.433**	0.353*	0.421**	0.385**	0.464**	0.293*	0.338*	0.083
U		0.338*	0.393**	0.376**	0.103	0.151	0.202	0.260
E			0.404**	0.545**	0.245	−0.060	0.129	0.092
J				0.457**	0.324*	0.239	0.303*	0.195
O					0.322*	0.154	0.301*	0.086
R						0.116	0.654**	0.246
SD							0.481**	0.135
SM								0.493**

*Significant at the 0.05 level.
**Significant at the 0.01 level.

Table 6. Spearman R for computer self-efficacy.

	CS	
	Pretest	Posttest
I	0.186	0.087
U	0.517**	0.334
E	0.679**	0.547**
J	0.463*	0.590**
O	0.428*	0.383
R	0.077	−0.078
SD	0.208	0.187
SM	0.254	0.321
SU	0.383	0.385

*Significant at the 0.05 level.
**Significant at the 0.01 level.

4.3. Hypothesis Testing

4.3.1. MANCOVA

Based on the MANCOVA design (Figure 3), the primary research hypothesis (Section 2.3) was transformed into the following three complementary statistical hypotheses:

– Main effect for F:

- *Null hypothesis H_0^F:* The means of I, U, E, J, O, R, SD', SM, and SU posttest measurements—adjusted for pretest measurements and CS measurements—are the same for variant B and variant C of SPEARMINT.
- *Alternative hypothesis H_1^F:* The means of I, U, E, J, O, R, SD', SM, and SU posttest measurements—adjusted for pretest measurements and CS measurements—are different for variant B and variant C of SPEARMINT.

– Main effect for T:

- *Null hypothesis H_0^T:* For variant B and variant C of SPEARMINT, the means of I, U, E, J, O, R, SD', SM, and SU—adjusted for CS—are the same for pretest and posttest measurements.
- *Alternative hypothesis H_1^T:* For variant B and variant C of SPEARMINT, the means of I, U, E, J, O, R, SD', SM, and SU—adjusted for CS—are different for pretest and posttest measurements.

– Interaction effect between F and T:

- *Null hypothesis H_0^{FT}:* There is no interaction between F and T.
- *Alternative hypothesis H_1^{FT}:* There is an interaction between F and T.

Table 7. MANCOVA—summary of all effects.

	Wilks' λ	Rao's R	df_1	df_2	p	Effect size
F	0.284	3.353	9	12	0.027*	71.6%
T	0.378	2.377	9	13	0.076	62.2%
FT	0.458	1.710	9	13	0.184	54.2%

*Significant at the 0.05 level.

The result of the MANCOVA is given in Table 7. Wilks' lambda (λ) and Rao's R are multivariate equivalents of the univariate F test. Rao's R—an approximation of the F statistic—is used for the test of significance.[6] Wilks' λ on the other hand is used to calculate the effect size as $1 - \lambda$. The effect size indicates the proportion of variance in the dependent variables, i.e., CASE tool acceptance, that can be attributed to the independent variable.

As can be seen, the main effect of F is significant at the 0.05 level. Thus, null hypothesis H_0^F can be rejected, and the alternative hypothesis H_1^F becomes valid. The main effect of T and the interaction effect between F and T are not significant at the 0.05 level. Thus, null hypotheses H_0^T and H_0^{FT} cannot be rejected. F alone explains 71.6% of the variance in CASE tool acceptance, which is an effect of medium size.

To evaluate the primary research hypothesis, only F is relevant. T had been introduced to consider the experiment's design as a training in using SPEARMINT, and to distinguish learning effects from the effect of F. Therefore, the primary research hypothesis, concretised as "the acceptance of SPEARMINT differs depending on whether it applies form CC1 or form CT4 to automatically assure hierarchical consistency," was confirmed by the experiment.

To evaluate the secondary hypotheses, i.e., whether predicted use was higher for variant B or variant C of SPEARMINT, the means of I, U, E, J, O, R, SD', SM, and SU were compared. As can be seen in Figure 11, the results are not uniform:

- Beliefs about consistency rules (i.e., SD', SM, and SU) are clearly more positive in the first group (F = B) than they are in the second group (F = C).

- While output quality (O) and result demonstrability (R) are higher in the second group, job relevance (J) is higher in the first group.

- Perceived usefulness (U) is higher in the first group, but perceived ease of use (E) is slightly higher in the second group. Intention to use (I) is about the same in both groups.

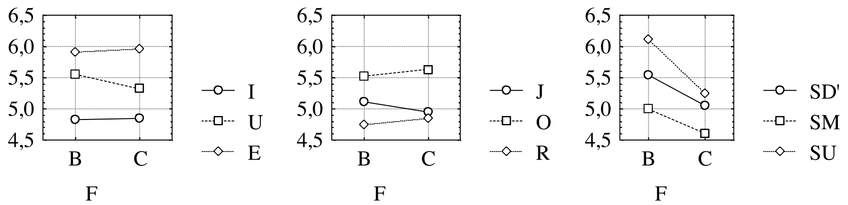


Figure 11. MANCOVA—means of CASE tool acceptance for main effect F.

Table 8. Alternative MANCOVA—summary of all effects.

	Wilks' λ	Rao's R	df_1	df_2	p	Effect size
F	0.449	2.454	6	12	0.088	55.1%
T	0.470	3.006	6	16	0.037*	53.0%
FT	0.621	1.625	6	16	0.204	37.9%

*Significant at the 0.05 level.

According to TAM, actual use of a CASE tool is predicted mainly by the intention to use it. Since intention to use is of similar size in both groups, the variant of SPEARMINT with the highest predicted use cannot be determined. The effect of form of automated consistency assurance (F) on CASE tool acceptance, whose existence has been demonstrated above, seems to be conflictive by nature.

4.3.2. *Alternative MANCOVA*

In the above analysis, beliefs about consistency rules play a central role. However, these were defined ad hoc, and nothing was known about the validity of the constructs and the reliability of the scales. Therefore, an alternative MANCOVA was performed, in which SD', SM, and SU were no longer treated as dependent variables but as covariates. For this purpose, only the pretest measurements of these variables were taken into account.

As can be seen in Table 8, the main effect of F was no longer significant at the 0.05 level, but the type I error remained low at 8.8%. However, a significant learning effect was found: Intention to use was clearly higher after the treatment (Figure 12).

Based on the alternative MANCOVA alone, an experimental confirmation of the primary research hypothesis wouldn't have been possible. Consequences thereof are discussed in Section 5.2 below.

4.3.3. *MANCOVA Assumptions*

The MANCOVA technique is based on the following assumptions:

– *Normal distribution:* It is assumed that the dependent variables are measured on at least an interval scale level. Moreover, they should be normally distributed within groups. However, the F test is remarkably robust to deviations from normality.

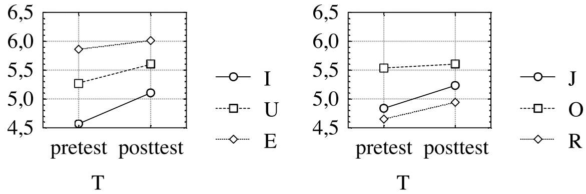


Figure 12. Alternative MANCOVA—means of CASE tool acceptance for main effect T.

– *Homogeneity of variances and covariances:* It is assumed that the variances of the dependent variables in the different groups of the design are identical, and that their inter-correlations (covariances) are homogeneous across the cells of the design. Not much is known about the robustness of Wilks' λ, the multivariate counterpart to the F test, to violations of this assumption.

Concerning this experiment, it must be stated that these assumptions were to some extent not fulfilled. But based on comments found in literature, it is acted on the assumption that the deviations range within acceptable limits, i.e., they can be neglected.

5. Discussion

5.1. Summary

The experimental results confirmed the primary hypothesis that the acceptance of SPEARMINT differed depending on whether it applied form CC1 (correct actively; don't interrupt) or form CT4 (tolerate actively; interrupt and display) to automatically assure hierarchical consistency. 71.6% of the variance in CASE tool acceptance could be attributed to the form of automated consistency assurance. Thus, sound empirical evidence has now been established for the theory that CASE tool acceptance depends on the methodology support it offers.

On the other hand, it was not possible to determine whether variant B (with form CC1) or variant C (with form CT4) of SPEARMINT would result in more frequent use. The effect of form of automated consistency assurance on CASE tool acceptance seems to be conflictive by nature, i.e., TAM variables are influenced in opposite directions. Solely beliefs about consistency rules were clearly more positive regarding form CC1 than regarding form CT4.

In addition, CADPRO's supposition, that beliefs about consistency rules play a major role in CASE tool acceptance, has been confirmed, even though the character of this role remains largely unclear.

5.2. Limitations

5.2.1. Internal Validity

History, *maturation*, and *testing* effects were controlled by the experimental design. If these effects had been active, it can be assumed that both groups were biased similarly. Thus, the findings are not affected. As it turned out, *experimental mortality* had been no problem either.

Statistical regression and *selection* effects were controlled by random assignment of subjects to groups. However, since blocking could not be applied, there was a small risk that randomisation had *not* created equivalent groups (Brooks et al., 2000b). Therefore, a

Mann–Whitney U test was performed and confirmed that there were no significant differences between both groups prior to the treatment.

For all TAM variables, *reliability of measures* can be considered sufficiently high since an approved instrument was used. However, nothing is currently known about the reliability of the measures that were used for beliefs about consistency rules. This poses a considerable threat to internal validity.

With respect to *statistical validity* it has to be stated that the sample was small and the assumptions of the MANCOVA were to some extent not fulfilled. But it is acted on the assumption that the deviations can be neglected since they range within acceptable limits. On the other hand, an alternative MANCOVA, in which beliefs about consistency rules were used as covariates only, did not confirm the research hypothesis.

The experiment had been designed as a practical training on a topic that was dealt with in theory by the class. However, after a presentation of the experiment's results given by the author, some subjects expressed the opinion that the experiment hadn't been sufficiently aligned with their learning targets. This belief was presumably caused by the presentation's form. However, if the belief had already existed during the experiment, the internal and external validity would have been affected. But the complaint maybe came from members of the second group (F = C), for which no learning effect could be found (see Section 5.2.2 below). Due to anonymous data collection, a closer analysis was yet not possible with hindsight.

To sum up, the internal validity is good, i.e., the findings argue for the primary research hypothesis. However, internal validity is challenged by the following factors:

1. The reliability of the measures that were used for beliefs about consistency rules is unknown.

2. The MANCOVA's statistical assumptions were to some extent not fulfilled.

3. An alternative MANCOVA, in which beliefs about consistency rules were used as covariates only, did not confirm the research hypothesis.

4. Subjects were possibly not sufficiently motivated.

5.2.2. External Validity

In parallel to the discussion above, *construct validity* and *instrumental validity* can be considered sufficiently high for all TAM variables. This is not true for beliefs about consistency rules; for them, construct validity and instrumental validity are unknown.

Whether *Hawthorne effects* or *pretest effects* restrict the generalisability of the findings depends on the extent to which the subjects' awareness and behaviour had been changed by participation in the experiment or the pretest, respectively. There is no reason to presume that these effects were different across groups. Therefore it's acted on the assumption that any difference between both groups would have been observed without them as well.

To avoid *sampling error*, it has to be considered that the subjects were students without experience in the method or the method's domain. The exercise to be solved was of pedagogical character and not very realistic. Generalising this situation to industrial practice is hardly possible. However, there is an important exception: the acquisition of a CASE tool while at the same time introducing a new method. In this case, professional users are in a learning situation that is limited in time and that is concluded by a purchase decision based on the CASE tool's acceptance.

The experiment was very specific: a single CASE tool (SPEARMINT) for a single method (descriptive process modelling) in a specific domain (process modelling) was considered; only three forms of automated consistency assurance (CP, CC1, CT4) were considered for three specific consistency rules related to hierarchical consistency. Nevertheless, it should be possible to disregard some of these aspects to generalise the findings of the experiment:

- There is no reason to presume that the observed effect of automated consistency assurance on CASE tool acceptance would only apply for descriptive process modelling and SPEARMINT. The effect might look different or might be weaker or stronger for other methods and CASE tools, but it's likely to exist as well.

- The consistency rules that were considered are related to hierarchical consistency. For this aspect, generalisability is more questionable. Literature suggests that hierarchical consistency is perceived as being complex and that active support is desired, whereas more passive support might be desired for local consistency. Further research is necessary to classify consistency rules accordingly before generalisations can be made.

5.3. Follow-up Survey

Since acceptance of SPEARMINT differed depending on the form of automated consistency assurance it applied, a small survey among more experienced users of SPEARMINT was conducted to collect some insight into the nature of this dependency.

The author performed structured interviews of about 30 minutes each with five SPEARMINT users. The subjects were confronted with several scenarios and had to select their *prefered form of automated consistency assurance* in each situation. The scenarios were based on a nested design, in which representative *consistency rules* of six available groups were either combined with three *task types* ("collect and sift information," "consolidate and structure model," and "correct and maintain model") or with two classes of *model size and complexity* ("small and simple" and "large and complex").

Using χ^2 tests for homogenity, it was found that subjects preferred different forms of automated consistency assurance, in particular depending on the character of the consistency rule and the task type. The model's size and complexity on the other hand didn't have any impact. A full description of the survey can be found in (Zettel, 2003).

6. Conclusion

This paper reports of a controlled experiment, in which two variants of the process modelling tool SPEARMINT were compared. A significant difference in CASE tool acceptance was found, which could be attributed to the form of automated consistency assurance that was applied for hierarchical consistency.

It was confirmed that beliefs about consistency rules play an important role in CASE tool acceptance: Without taking them into account, the experiment's result wouldn't have been significant. This also poses a considerable threat to the validity of the findings, since neither construct validity nor instrumental validity and reliability have been investigated so far. Therefore, this should be made up for prior to performing further empirical studies.

The experimental design was based on the circumstance that little time was available to train the subjects and perform the treatment. As a consequence, the observed effect of the experimental variable was rather small. In addition, a very complex statistical analysis technique had to be applied, whose assumptions were to some extent not fulfilled. All this poses a risk to the experiment's internal validity. For replications of the experiment, it should be checked whether more time is available and the MANCOVA may be safely replaced by a Mann–Whitney U test.[7]

To explore the nature of the relationship further, a small survey among experienced SPEARMINT users was performed. It was found that subjects preferred different forms of automated consistency assurance depending on individual, consistency rule, and task characteristics. Therefore, future studies using CASE tools with configurable consistency assurance should be performed to develop a general classification of consistency rules based on empirical data.

From a more practical perspective, the findings presented in this paper stress that CASE tool users need *varying* forms of automated consistency assurance to yield optimal task-technology fit. Thus, vendors should provide CASE tools with *adaptable* methodology support, which allow their users to fit them to the tasks at hand.

Appendix

Variables and Scales

Table 9 shows the variables of the controlled experiment. The control variables and the dependent variables had to be measured using two questionnaires: O_1 and O_2. For all TAM variables, well-known scales consisting of two to four items were taken from literature (Table 10). For beliefs about consistency rules, new scales were defined consisting of one item each (Table 11). These items were defined based on matching TAM variables. According to a recommendation of Davis and Venkatesh (Davis and Venkatesh, 1996), items were grouped by variables instead of ordering them randomly. Subjects could respond to the items using seven point Likert scales, ranging from disagree (coded as 1) to agree (coded as 7).

Table 9. Variables.

	Variable	Range	Measurement
F	Form of automated consistency assurance	A = CP, B = CC1, C = CT4	implicit
ES	Prior experience with SPEARMINT	1 = no, 2 = yes	O_1
CS	Computer self-efficacy	1 = disagree . . . 7 = agree	O_1
I	Intention to use	1 = disagree . . . 7 = agree	O_2
U	Perceived usefulness	1 = disagree . . . 7 = agree	O_2
E	Perceived ease of use	1 = disagree . . . 7 = agree	O_2
J	Job relevance	1 = disagree . . . 7 = agree	O_2
O	Output quality	1 = disagree . . . 7 = agree	O_2
R	Result demonstrability	1 = disagree . . . 7 = agree	O_2
SD	Detectability of inconsistencies	1 = disagree . . . 7 = agree	O_2
SM	Meaningfulness of consistency rules	1 = disagree . . . 7 = agree	O_2
SU	Usefulness of consistency rules	1 = disagree . . . 7 = agree	O_2

In most of the statistical analyses, the inverse variable SD’ (SD’ = 8 − SD) was used instead of SD.

Table 10. Scales—TAM variables.

Var.	Items
I	Assuming I have access to SPEARMINT, I intend to use it.
	Given that I have access to SPEARMINT, I predict that I would use it.
U	Using SPEARMINT improves my performance in process modelling.
	Using SPEARMINT in process modelling increases my productivity.
	Using SPEARMINT enhances my effectiveness in process modelling.
	I find SPEARMINT to be useful in process modelling.
E	My interaction with SPEARMINT is clear and understandable.
	Interacting with SPEARMINT does not require a lot of my mental effort.
	I find SPEARMINT to be easy to use.
	I find it easy to get SPEARMINT to do what I want to do.
J	In process modelling, usage of SPEARMINT is important.
	In process modelling, usage of SPEARMINT is relevant.
O	The quality of the output I get rom SPEARMINT is high.
	I have no problem with the quality of SPEARMINT’s output.
R	I have no difficulty telling others about the results of using SPEARMINT.
	I believe I could communicate to others the consequences of using SPEARMINT.
	The results of using SPEARMINT are apparent to me.
	I would have difficulty explaining why using SPEARMINT may or may not be beneficial.
CS	My interaction with a computer is clear and understandable.
	Interacting with a computer does not require a lot of my mental effort.
	I find a computer to be easy to use.
	I find it easy to get a computer to do what I want it to do.

Table 11. Scales—other variables.

Var.	Items
ES	Did you use SPEARMINT before?
SD	I believe I could find all inconsistencies in a process model without using SPEARMINT.
SM	I would have no difficulty explaining why the consistency rules are meaningful.
SU	I find the consistency rules to be useful in process modelling.

In case of multi-item scales, the total value was calculated as arithmetic mean of the single values.[8] This allowed to compensate missing values. For the analysis, Likert scales were treated as interval scales.

Acknowledgments

I thank Prof. Dieter Rombach from Fraunhofer IESE for supervising my PhD research, on which this article is based. I also thank Peter Rösch, Louise Scott, and Jürgen Münch for their support in various stages of my research. Further, I thank Prof. Ross Jeffery and Andrew Brooks for their helpful feedback on early versions of this article. And finally, I thank the anonymous reviewers for their constructive comments.

This work was financially supported by Stiftung Rheinland–Pfalz für Innovation (project no. 450) and Fraunhofer IESE.

Notes

1. In software engineering, *methods* usually deal with *models*. Which models to create and how to structure their content is defined by the method's *metamodel* given or example as UML class diagram(s). For models to be well-formed, they must adhere to the method's *consistency rules*, given for example as OCL constraints. The appearance of models is defined by the method's *notation* providing graphical or textual symbols or all concepts. Finally, the method's *process* defines the procedural steps to be followed during modelling with more or less precision. The term *methodology support* has been established in literature to denote a CASE tool's functionality that assists its users in applying one or more software engineering methods.
2. SPEARMINT is a registered trademark of Fraunhofer IESE.
3. A demo version of SPEARMINT is available free of charge from Fraunhofer IESE's web site at `http://www.iese.fhg.de/Spearmint_EPG`.
4. Recently, TAM and our other theoretical models of individual acceptance of information technology *tools* have been examined in the domain of individual aceptance of *methodologies* as well (Riemenschneider et al., 2002).
5. In general, characterisation of the methodology support provided by a CASE tool takes place by specifying for every consistency rule the form of automated consistency assurance that is applied.
6. Use the two degrees of freedom df_1 and df_2 to look up the R value in an F table.
7. For the experiment at hand, univariate analysis techniques were applied as well:

   – A Mann–Whitney U test was performed for between-group differences in posttest CASE tool acceptance. A significant difference was found only for usefulness of consistency rules; but this difference was already close to significance during pretest.

– Another Mann–Whitney U test was performed for between-group differences in increases in CASE tool acceptance from pretest to posttest. The only significant difference found here was in output quality increase.

– To test whether the subjects' CASE tool acceptance changed from pretest to posttest, a Wilcoxon's matched pairs test was performed. Significant changes were found only in the first group (F = B), namely increases in intention to use, perceived usefulness, perceived ease of use, job relevance, and result demonstrability.

As it had been expected, it was impossible to confirm the research hypothesis using univariate analyses techniques. But the results of the Wilcoxon's matched pairs test indicate at least an obvious difference between both groups: Significant changes of CASE tool acceptance were observable in the first group only. However, the multivariate approach was still necessary to confirm this.

8. According to (Kerlinger, 1986, p. 454), the total value of a multi-item Likert scale can be calculated by summing up the single values or by computing the arithmetic mean.

References

Adams, D. A., Nelson, R. R., and Todd, P. A. 1992. Perceived usefulness, ease of use, and usage of information technology: A replication. *MIS Quarterly* 16(2): 227–247.

Becker-Kornstaedt, U., Hamann, D., Kempkens, R., Rösch, P., Verlage, M., Webby, R., and Zettel, J. 1999. Support for the process engineer: The spearmint approach to software process definition and process guidance. In: M. Jarke and A. Oberweis (eds.), *Proceedings: Advanced Information Systems Engineering: 11th International Conference, CAiSE'99, Heidelberg, Germany, June 14–18, 1999*, Springer, pp. 119–133.

Brooks, A., and Sobczak, A. 1999. CASE tool support for high-level refactorings: Keystroke-level model comparisons. In *5th Conference on Smalltalk and Java in Industry and Education (STJA 99), 28–30 September 1999*, Erfurt, Germany.

Brooks, A., Takada, S., and Scott, L. 1999. Strongly formative pilot studies on constraints in early life-cycle work. In *Proceedings of the Sixth Asia-Pacific Software Engineering Conference '99 (APSEC'99), 7–10 December, 1999*, IEEE Computer Society, Takamatsu, Japan, pp. 614–621.

Brooks, A., Campbell, B., and Scott, L. 2000a. The human–computer interface explanation: A correspondence on Jankowski's paper on methodological support for structured analysis. *Empirical Software Engineering* 5(1): 69–71.

Brooks, A., Utbult, F., Mulligan, C., and Jeffery, R. 2000b. Early lifecycle work: Influence of individual characteristics, methodological constraints, and interface constraints. *Empirical Software Engineering* 5(3): 269–285.

Brooks, A., and Scott, L. 2001. Constraints in CASE tools: Results from curiosity driven research. In *Proceedings of the 2001 Australian Software Engineering Conference*, 27–28 August 2001, IEEE Computer Society, Canberra, Australia. pp. 285–293. Also published as CAESAR Report No.98/7.

Card, D. N., McGarry, F. E., and Page, G. T. 1987. Evaluating software engineering technologies. *IEEE Transactions on Software Engineering* SE-13(7): 845–851.

Chau, P. Y. K. 1996. An empirical investigation on factors affecting the acceptance of CASE by systems developers. *Information & Management* 30(6): 269–280.

Davis, F. D. 1986. A technology acceptance model for empirically testing new end-user information systems: Theory and results. Ph.D. thesis, Massachusetts Institute of Technology, Sloan School of Management, Cambridge.

Davis, F. D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* 13(3): 318–340.

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. 1989. User acceptance of computer technology: A comparison of two theoretical models. *Management Science* 35(8): 982–1003.
Davis, F. D. 1993. User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man–Machine Studies* 38: 475–487.
Davis, F. D., and Venkatesh, V. 1996. A critical assessment of potential measurement biases in the technology acceptance model: Three experiments. *International Journal of Human–Computer Studies* 45: 19–45.
Day, D. L. 1996. User responses to constraints in computerized design tools: An extended abstract. *ACM SIGSOFT Software Engineering Notes* 21(5): 47–50.
Day, D., Ahuja, M., and Scott, L. 1997. Constraints in design engineering: A report on research in progress. Technical Report 97/5, The University of New South Wales, Centre for Advanced Empirical Software Research. Published in: *Proceedings of the 8th Australian Conference on Information Systems*, pp. 509–516.
DeLone, W. H., and McLean, E. R. 1992. Information systems success: The quest for the dependent variable. *Information Systems Research* 3(1): 60–95.
Fichman, R. G., and Kemerer, C. F. 1999. The illusory diffusion of innovation: An examination of assimilation gaps. *Information Systems Research* 10(3): 255–275.
Fischer, G., Lemke, A. C., Mastaglio, T., and Morch, A. I. 1991. Critics: An emerging approach to knowledge-based human–computer interaction. *International Journal of Man–Machine Studies* 35(5): 695–721.
Fishbein, M., and Ajzen, I. 1975. *Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research Addison–Wesley Series in Social Psychology*. Reading: Addison–Wesley.
Fuggetta, A. 1993. A classiffication of CASE technology. *IEEE Computer* 26(12): 25–38.
Glass, R. L. 1999. The realities of software technology—Payoffs. *Communications of the ACM* 42(2): 74–79.
Goodhue, D. L. 1995. Understanding user evaluations of information systems. *Management Science* 41(12): 1827–1844.
Goodhue, D. L., and Thompson, R. L. 1995. Task-technology fit and individual performance. *MIS Quarterly* 19(2): 213–236.
Henderson, J. C., and Cooprider, J. G. 1990. Dimensions of I/S planning and design aids: A functional model of CASE technology. *Information Systems Research* 1(3): 227–254.
Henderson, R., Rickwood, D., and Roberts, P. 1998. The beta test of an electronic supermarket. *Interacting with Computers* 10(4): 385–399.
Iivari, J. 1996. Why are CASE tools not used? *Communications of the ACM* 39(10): 94–103.
ISO 12207. 1995. Information technology—Software life cycle processes. International Standard ISO/IEC 12207.
Jankowski, D. 1997. Computer-aided systems engineering methodology support and its effect on the output of structured analysis. *Empirical Software Engineering* 2(1): 11–38.
Jarzabek, S., and Huang, R. 1998. The case for user-centered CASE tools. *Communications of the ACM* 41(8): 93–99.
Jeffery, D. R., and Offen, R. J. (eds.). 1999. CADPRO—Experiments in CASE tool use and constraint conditions: In *Proceedings of the 1999 CADPRO Workshop, 25th September 1999*, The University of New South Wales, Centre for Advanced Empirical Software Research.
Kerlinger, F. N. 1986. *Foundations of Behavioral Research*. 3rd edition. Fort Worth: Harcourt Brace Jovanovich College Publishers.
Laitenberger, O., and Dreyer, H. M. 1998. Evaluating the usefulness and the ease of use of a web-based inspection data collection tool. Technical Report ISERN-98-13, Fraunhofer Institute for Experimental Software Engineering, Germany. Also published as IESE-Report No. 027.98/E.
Mathieson, K. 1991. Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research* 2(3): 173–191.
Morris, M. G., and Dillon, A. 1997. How user perceptions influence software use. *IEEE Software* 14(4): 58–65.
Riemenschneider, C. K., Hardgrave, B. C., and Davis, F. D. 2002. Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Transactions on Software Engineering* 28(12): 1135–1145.
Robbins, J. E., and Redmiles, D. F. 1999. Cognitive support, UML adherence, and XMI interchange in Argo/UML. In: J. Gray, J. Harvey, A. Liu, and L. Scott, (eds.). *Proceedings of The First International Symposium on Constructing Software Engineering Tools (CoSET'99), 17–18 May, 1999, Los Angeles, USA*. Mawson

Lakes, SA 5095, Australia, School of Computer and Information Science, University of South Australia, pp. 61–70.

Roberts, P., and Henderson, R. 2000. Information technology acceptance in a sample of government employees: A test of the technology acceptance model. *Interacting with Computers* 12(5): 427–443.

Segars, A. H., and Grover, V. 1993. Re-examining perceived ease of use and usefulness: A confirmatory factor analysis. *MIS Quarterly* 17(4): 517–525.

Silverman, B. G., and Mehzer, T. 1997. A study of strategies for computerized critiquing of programmers. *Empirical Software Engineering* 2(3): 339–359.

Szajna, B. 1994. Software evaluation and choice: Predictive validation of the technology acceptance instrument. *MIS Quarterly* 18: 319–324.

Taylor, S., and Todd, P. A. 1995. Understanding information technology usage: A test of competing models. *Information Systems Research* 6(2): 144–176.

Venkatesh, V., and Davis, F. D. 1996. A model of the antecedents of perceived ease of use: Development and test. *Decision Sciences* 27(3): 451–481.

Venkatesh, V., and Davis, F. D. 2000. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science* 46(2): 186–204.

Vessey, I., Jarvenpaa, S. L., and Tractinsky, N. 1992. Evaluation of vendor products: CASE tools as methodological companions. *Communications of the ACM* 35(4): 90–105.

Zettel, J. 2001. Methodological support for descriptive software process modeling: Consistency rules in SPEARMINT 5. Technical Report IESE-Report 014.01/E, Fraunhofer Institute for Experimental Software Engineering.

Zettel, J. 2003. *Anpassbare Methodenassistenz in CASE-Werkzeugen*, Vol. 13 of *PhD Theses in Experimental Software Engineering*. Stuttgart: Fraunhofer IRB Verlag. PhD Thesis, University of Kaiserslautern, Department of Computer Science; ISBN 3-8167-6284-0.

Jörg Zettel received his M.S. (Diplom) and PhD degrees in computer science (Informatik) from the University of Kaiserslautern, Germany, in 1995 and 2003, respectively. From 1996 to 2003, he was research assistant at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern. Since 2003, he is working as an analyst for the Hessische Zentrale für Datenverarbeitung (HZD) in Wiesbaden, Germany, where he became project leader of the corporate improvement program for software development in 2004. His current research interests include software processes, computer-aided software engineering, component-based software engineering, and agile software development.