Comparing Use Case Writing Guidelines

Karl Cox, Keith Phalp, Martin Shepperd

Empirical Software Engineering Research Group
Dept of Computing
Bournemouth University, UK
{coxk, kphalp, mshepper}@bmth.ac.uk

Abstract. Use cases are a widely used technique for requirements specification as part of the Unified Modelling Language (UML). However, use cases rely predominantly upon natural language. For this reason, the CREWS research group has proposed guidelines to assist in writing use cases. Various research groups, including ourselves, have found that writing guidelines (CREWS) help. However, our experience with students is that the CREWS guidelines are a little unwieldy, and can be difficult to apply. In this paper we propose some simplified (CP) use case guidelines, particularly in terms of admissible structures. We then describe a pilot experiment to explore whether the simplifications result in any loss of use case quality. Our initial results suggest that our simpler guidelines perform at least as well as the CREWS guidelines. Consequently, we suggest that, given these promising findings, further empirical studies, particularly industrial case studies, be conducted to confirm whether our simplified approach warrants industrial adoption.
Keywords: use cases, experiment, guidelines

1 Introduction

With the emergence of the UML [1, 2], use cases [3] have become an accepted requirements tool in object-oriented software engineering. However, although the use case diagram has received much attention, little advice has been given on how use cases should be written. To address this issue, Cockburn [4] provides a wealth of information on writing use cases; indeed, our own CP Writing Rules borrow some of these ideas. The EU funded research project CREWS (Co-operative Requirements Engineering With Scenarios) has produced a great deal of excellent work on use cases and scenarios. Ben Achour [5, 6] and colleagues [7] have investigated and developed a comprehensive set of writing guidelines for use cases in an effort to answer industrial concerns [8]. We consider the CREWS Guidelines to be an important contribution to use cases. Indeed, our previous work [9] with the CREWS guidelines has shown that guidance does improve the completeness of the use case description. Good as the CREWS Guidelines are, we believe that they can be overly complex to

apply. Therefore, we propose an alternative simpler set of guidelines: the CP Writing Rules, which are discussed in the next section. Section 3 considers how we judge use case descriptions, section 4 describes the experimental design, section 5 reports on the experimental results, section 6 draws conclusions and section 7 makes some observations about the experiment and about further work.

2 The CP Use Case Writing Rules

For reasons of space, we cannot show the complete CREWS guidelines here. However, they can be found, with examples of their application, at [10]. As stated, use cases written under guidance tend to be better than without. However, our students have found the CREWS guidelines are a little unwieldy and difficult to apply. We have developed the CP Rules as an alternative simpler set of guidelines. We do not set out to better CREWS but hope to achieve similar results with simpler guidelines. The CP Writing Rules consists of two parts: Style and Structure.

2.1 CP Style Rules

Style 1: Each sentence in the description should be on a new, numbered line. Alternatives and exceptions should be described in a section below the main description and the sentence numbers should agree.
Style 2: Avoid pronouns if there is more than one actor.
Style 3: No adverbs or adjectives.
Style 4: Avoid negatives.
Style 5: Give explanations if necessary.
Style 6: All verbs are in present tense format.
Style 7: There should be logical coherence throughout the description.
Style 8: When an action occurs there should be a meaningful response to that action.

Students have found the CREWS Style Guidelines to be very detailed and take time to read. As such, we have attempted to simplify them, although the CP Style Rules draw much from the CREWS Style Guidelines. There are, though, some minor differences. CP Style 5 allows the writer to expand upon details if necessary, and to adopt different levels of granularity. In CP Style 7, we recognise that local coherence, and to a degree, global coherence, improves text comprehension both in terms of meaning and time taken to read [11]. Making the writer aware of this should induce more readable use cases. CP Style 8 addresses logical flow. This refers to the *question->reply to question* principle of text comprehension [12]. Use cases are read by many stakeholders, so should be as easy to read as possible. This rule encourages the writer to consider this fact.

2.2 CP Structure Rules

The CP Structure Rules have only four structures, half the number of the comparable CREWS Content Guidelines.
Structure 1: Subject verb object.
Structure 2: Subject verb object prepositional phrase.
Structure 3: Subject passive.
Structure 4: Underline other use case names.

Structure 1 is the same as CREWS CG5. After a survey of over 150 published use cases or written by the authors' students, it was found that this structure was used most often (18%) to construct sentences. The total number of events in the surveyed use cases is 1,913.

In the survey the structure next most common is: *subject verb object preposition object* (6% occurrence). This structure is recommended by Graham [13] and is similar to Cockburn's *subject verb object prepositional phrase* [4]. We take Cockburn's structure as CP Structure 2 because it allows for more word variation than the CREWS Content Guidelines 1 to 3. Figure 1 (left) shows the percentage of CP Structure (CP Str) Rules that we found to be applied in the survey. CP Structure 3 (subject passive) has 2% occurrence.

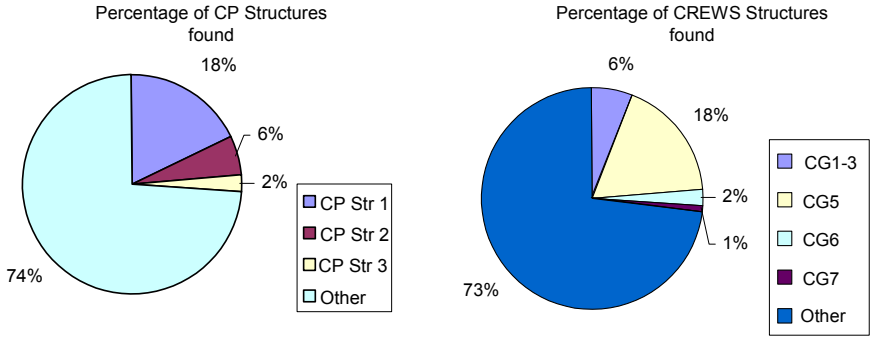

Fig. 1. Percentage CP Use Case Structure Rules (left) and CREWS Use Case Guidelines (right) found in the survey.

Figure 1 (right) also shows equivalent percentage of CREWS Content Guidelines (CG). Apart from CG5, (equivalent of CP Structure 1), and CGs1-3, (equivalent of CP Structure 2), the other CREWS CGs that appear in the survey are CG6: 'If' <alternative assumption> 'then' <action>, which has an occurrence in 2%. We take the guideline to be any form of an if… then… statement. Next is CG7: 'Loop' <repetition condition> 'do' <action>, 1% occurrence. CG4 and CG8 do not occur.

Figure 1 (left and right) both suggest much scope for either set of guidelines as c.75% of use cases fall outside of admissible structures i.e. writers are pretty unconstrained.

Having constructed an alternative set of guidelines we then wished to judge their efficacy by comparing their effect with those of the CREWS guidelines. Our previous work had already confirmed that writers using the CREWS guidelines produced more complete use cases than without guidelines, and hence we viewed the CREWS set as an ambitious baseline. However, in order to be able to examine the effectiveness of guidelines, we needed a sensible way to judge (or mark) the use cases produced. This is the focus of the following section.

3 Judging Use Case Descriptions

One approach to the evaluation of use cases is to count the number of times a rule or guideline has been applied. Indeed, if one believes in the efficacy of the guidelines then there is clearly some merit in judging the extent of its application. For example, we might count attributes such as the number of times a specific grammatical structure is employed. This gives a quantitative assessment of the use case. The risk is that judging the improvement of a use case in this way could become a self-fulfilling prophecy. A high count does not necessarily tell us that the use case is improved [9]. It would be better if there were another (independent) way to judge use cases. Since a primary goal of use cases is communication between stakeholders, it would seem that one way to assess a use case would be to decide how communicable it is. We suggest some important factors to consider in their assessment. The subjectivity of this approach is clearly a weakness. However, as with many aspects of software engineering, the qualitative assessment of documents by application of heuristics appears to offer a suitable way to assess use cases. "Question asking" is recognised as important in the establishment of the completeness to individual elements of the use case model [14]. We focus our question set on the description itself.

3.1 Use Case Heuristics

Here we describe four characteristics of the use case description, which are all aspects of its ability to communicate. These (four C's) are: Coverage, Coherence of logic and its readability, Consistency of structure and Consideration of alternatives.

Coverage

- Completes: does the use case terminate or does the path get caught in a loop?
- Rational: does the use case provide a rational answer to the problem?
- Span: does the use case contain all that is required to answer the problem?

- Scope: does the use case contain detail only relevant to the problem statement or does the description provide extra unnecessary information?

Coherence
- Logical order: does the use case follow a logical path? Is this path clear?
- Logical coherence [11]: is there coherence through the use case both locally and globally, thus making the use case easier and quicker to read?
- Abstraction: is the use case at a consistent level of abstraction throughout? Mixing abstraction levels will cause difficulty in understanding.

Consistent Structure
- Variations: are alternative paths excluded from the main flow? Inclusion of alternative paths in the main flow reduces readability.
- Consistent Grammar: is present tense used throughout? Are adverbs, adjectives, pronouns and negatives avoided?
- Sequence: is numbering of events in the main flow consistent?

Consideration of Alternatives
- Separation: is there a separate section for alternative/exceptional paths to the main flow?
- Viable: are the alternatives viable?
- Numbering: do the alternative numberings match the numbers in the main flow?

4. The Experimental Design

The paper describes a pilot experiment that compares the CP Rules with the CREWS Guidelines. As such, we cannot generalise results because of the small scale of the study and the artificial nature of the tasks but we can consider the results as guides to further, larger studies. The experiment sets out to test the quality of the use cases written with the aid of CREWS guidelines against the CP rules. The experiment is evaluated in terms of time taken to write the use case, the length of the use case, the count of usage of the guidelines or rules and the comprehensibility of the use case.

For the main body of the experiment, there were four groups of six subjects each (24 subjects in all). The subjects were post-graduate students attending a Masters course in software engineering. The subjects were given a half-day seminar on use cases one week before the experiment. A variable outside our control is how much further knowledge the subjects gained on writing use cases in the week between the seminar and the experiment. Table 1 shows the experimental groups and the tasks for phases 3 and 4 in the experiment. The entire experiment consisted of 5 phases:

- Phase 1: experience questionnaire
- Phase 2: pre-experimental task - to help provide an equal distribution of ability and experience across the groups (see [10] for questionnaire and task) we used randomised blocking.
- Phase 3: write the use cases. The subjects were given instructions, a set of guidelines and task - ATM or Retail (see [10]). The subjects were given one hour to complete the task, and were then asked to complete a short questionnaire about the writing guidelines they had used [10].
- Phase 4: subjects read a use case written with the aid of the guidelines they themselves employed, but on a different problem (see table 1). The subjects were then asked to complete a (comprehension) questionnaire about the use case they had read [10]. The subjects had 30 minutes to complete this phase.
- Phase 5: a qualitative feedback session for all groups in which the whole experiment was discussed.

Table 1. Groups and tasks

Group		Phase 3: Write UC	Phase 4: Read UC
CP	A	ATM	Retail (B)
	B	Retail	ATM (A)
CREWS	C	ATM	Retail (D)
	D	Retail	ATM (C)

4.1 Experimental Hypotheses

Because of the small sample size we cannot generalise our results. However, we propose hypotheses that relate to a (surrogate) measure of use case quality:

H1: The CREWS Guidelines produce significantly better use cases than the CP Rules in terms of the time it takes to write the use cases and their length.

H2: The CREWS Guidelines produce significantly better use cases than the CP Rules in terms of the number of events that correctly implement the rules or guidelines.

H3: The CREWS Guidelines produce significantly better use cases than the CP Rules in terms of comprehensibility of the use cases.

There is a null hypothesis for each alternative that states there is no significant difference between CREWS and CP.

5 Experimental Results

We now examine the experimental results in terms of the three alternative hypotheses. Section 5.1 discusses Hypothesis 1 (efficiency), section 5.2 Hypothesis 2 (application

of guidelines) and section 5.3 Hypothesis 3 (comprehension). The complete statistical sets for the experiment can be found at [10].

5.1 Hypothesis 1: Efficiency

Hypothesis 1 tests the time taken to write the complete use case and the length of the *main flow* of each use case. Average times across the groups show the CREWS groups (C and D) are written 5.5% faster than the CP groups. Yet, the CP use cases are 23% shorter than the CREWS use cases. However, when we performed 1-tailed, unequal variance t-tests, the results show there is no significant difference between the two sets of guidelines. We take $\alpha = 0.05$. CP performs as well as CREWS (see table 2).

Table 2. Comparing Time and Length

Time				Time				Length				Length			
A1	60	C1	55	B1	56	D1	60	A1	11	C1	30	B1	13	D1	16
A2	55	C2	55	B2	60	D2	50	A2	26	C2	27	B2	6	D2	13
A3	60	C3	57	B3	45	D3	45	A3	15	C3	37	B3	28	D3	33
A4	45	C4	56	B4	60	D4	50	A4	34	C4	19	B4	5	D4	22
A5	50	C5	30	B5	60	D5	52	A5	35	C5	20	B5	15	D5	13
A6	53	C6	60	B6	55	D6	48	A6	12	C6	17	B6	24	D6	44
p = 0.38				p = 0.07				p = 0.31				p = 0.11			

Hence, hypothesis 1 is rejected. However, our results lead us to consider that these two attributes, length and duration, should have been considered separately. It does appear that CP rules produce smaller (more compact) use cases. Cockburn [4], for instance, takes shortness of use case to be a good thing, though our experiences in industry do not reflect this. In any case, without normalising against some other factor (e.g. completeness) it is not clear that this is necessarily an advantage.

5.2 Hypothesis 2: Count of Guidelines/Rules Applied

Hypothesis 2 is difficult to test because not all of the CP Rules match all the CREWS Guidelines. Because of lack of space we cannot go into detail but state that overall both CP and CREWS Style guidelines fare similarly well. The reader is referred to [10] for more information.

Comparing CP Structure and CREWS Content. Only two meaningful comparisons can be made: CP Structure 1 compared against CREWS Content Guideline 5 and CP Structure 2 compared against CREWS CGs 1, 2 and 3 (table 3).

The values represent the mean usage of the guideline in each use case description. We applied 1-tailed, unequal variance t-tests to find any significant difference.

Table 3. Comparing CP Structure 1 and 2 against CREWS CG5 and CG1-3

CP Structure 1		CREWS CG5		CP Structure 2		CREWS CG1-3	
A1	74.07	C1	23.33	A1	7.41	C1	10
A2	41.94	C2	65.96	A2	6.45	C2	0
A3	84	C3	54.55	A3	6	C3	6.07
A4	57.78	C4	8	A4	0	C4	4
A5	56.52	C5	21.74	A5	8.7	C5	0
A6	29.41	C6	11.76	A6	11.76	C6	0
B1	0	D1	40	B1	15.38	D1	15
B2	20	D2	26.32	B2	10	D2	21.05
B3	28.95	D3	34.04	B3	5.26	D3	0
B4	0	D4	22.73	B4	0	D4	0
B5	43.33	D5	61.54	B5	36.67	D5	3.85
B6	12	D6	22.73	B6	12	D6	2.27
Probability				Probability			
AC against BD		p = 0.32		AC against BD		p = 0.09	
A against C		p = 0.03		A against C		p = 0.09	
B against D		p = 0.05		B against D		p = 0.18	

CP Structure Rule 1 (subject verb object) is used to structure 65.5% of events for group A against 40% usage of CREWS Content Guideline 5 for group C, and is significantly different (p=0.03). Comparing group B against group D is weakly significant (p=0.05). The reduced usage of Structure 1 and CG5 by groups B and D possibly reflects on the more complex use case path of the Retail task. There is no significant difference in usage of CP Structure 2 and CREWS Content Guidelines 1-3.

We can state that there is no statistically significant difference overall between the CP Structure Rules and the CREWS Content Guidelines. However, there is a statistical difference in favour of CP Structure 1 against CREWS CG5 especially for group A against C and weakly for B against D. We reject the hypothesis H2, stating that CP does as well as CREWS.

5.3 Comprehension

Marking to assess comprehension. Judging use cases required us to allocate marks to each use case in terms of a number of heuristics. These heuristics (section 3) allow us to assess use cases in a way that is independent of counts of usage. In fact, this heuristic judgement is typical of how one would assess many software artefacts. For example, we typically judge designs in a similar way. We applied t-tests (1-tailed, unequal variance) to find the statistical differences. Table 4 shows the marks awarded for groups A and C and the significance of each heuristic.

When comparing groups B and D (table 5), we find that these two groups of use cases are equally comprehensible and that there is no significant difference in any of the four question sets. However, when comparing ATM solutions (table 4), Group A's use cases are strongly significantly better than Group C's use cases in terms of coverage (p=0.004), significant in coherence (p=0.04), and weakly significant in

consistent structure (p=0.05). There is no significant difference in alternatives. To summarise, it does appear that use cases written with the CP Rules are at least as comprehensible as use cases written with the CREWS guidelines and even more so in the case of group A (coverage, coherence and structure). Hence, we reject hypothesis 3, with the observation that problem type appears to have an impact on the performance of the CP rules.

Table 4. Comparing groups A and C (ATM Task)

Coverage (30%)				Coherence (30%)				Structure (30%)				Alternatives (10%)			
A1	10	C1	10	A1	20	C1	15	A1	15	C1	10	A1	5	C1	0
A2	20	C2	10	A2	15	C2	17	A2	15	C2	18	A2	3	C2	8
A3	10	C3	10	A3	20	C3	20	A3	20	C3	20	A3	8	C3	5
A4	20	C4	0	A4	20	C4	5	A4	20	C4	10	A4	3	C4	5
A5	20	C5	5	A5	16	C5	5	A5	17	C5	10	A5	3	C5	0
A6	18	C6	10	A6	15	C6	10	A6	15	C6	10	A6	5	C6	0
p = 0.004				p = 0.04				p = 0.05				p = 0.19			

Table 5. Comparing groups B and D (Retail Task)

Coverage (30%)				Coherence (30%)				Structure (30%)				Alternatives (10%)			
B1	10	D1	15	B1	7	D1	15	B1	5	D1	15	B1	0	D1	2
B2	10	D2	15	B2	15	D2	10	B2	18	D2	12	B2	3	D2	6
B3	10	D3	20	B3	10	D3	10	B3	15	D3	10	B3	6	D3	3
B4	3	D4	10	B4	7	D4	7	B4	10	D4	5	B4	0	D4	0
B5	5	D5	5	B5	10	D5	15	B5	15	D5	15	B5	5	D5	4
B6	20	D6	5	B6	15	D6	5	B6	15	D6	10	B6	0	D6	0
p = 0.29				p = 0.44				p = 0.24				p = 0.46			

Groups B, C and D have comparable scores and CREWS have consistency over both tasks. As the CP Rules appear to be more successful when applied to the ATM task than when applied to the Retail use case, we might focus sets of guidelines more closely, to consider the possibility of problem frames for such use cases and therefore ways in which they ought to be written.

Reading use cases. After reading the use case, students were asked to answer questions about the logic and plausibility of the use case that they had been given (phase 4 of the experiment). For example, one reader suggests why A3's use case is not completely plausible: "The customer never enters the PIN number; therefore, all of the transactions cannot be completed." The use case itself states: "System validates

card." The underlined text (CP Structure 4) indicates this is a link to another use case. Whether this use case asks for a PIN number is impossible to know.

There is a growing acceptance among use case writers of applying Constantine and Lockwood's [15] "essential use case" idea. This would disallow the usage of terms such as "card" and "PIN" in a use case because they might force premature interface design. The granularity of use case description is specifically considered in our use case heuristics on coherence (section 3.1) when we look at abstraction levels within descriptions. The idea of granularity of use cases is considered in CP Style 5, which is explained in detail at [10]. We do think that avoidance of interface design is an ideal but our industrial experiences in re-engineering of an online brokerage system shows this not to be always practical.

Use Case Logic. There is general agreement between the subjects and the authors as to the logic of the use cases, with a few exceptions (see [10]). We show three examples:

1. Use case C1 has many alternative paths in the main flow. These alternatives should form the 'happy day scenario' path through the use case. We are confronted with barriers to the flow of the use case, with alternatives acting as the clearest route through. Some of the alternatives also fail to offer an escape, e.g.: "If there is not enough money in the customer's account then the ATM displays the 'Options' screen." We do not know where to go from here.
2. In use case C2 the Customer twice opts to withdraw money but the ATM does not offer the money. This might conform to Constantine and Lockwood's essential use case [15] but, as readers, we do not know if the money has been offered and taken, which questions the success of the use case.
3. Use case C4 gets stuck in a loop. Interestingly, the subject who read this use case states: "There is… no return card option… Customer stuck there forever." This indicates there are problems with the logical flow of the use case. But in general comment on the use case, the reader then comments, "The statements are well written and follow a logical progression." Despite this contradiction on the part of the subject, we consider the use case illogical.

We found logic problems in all of the CREWS use cases. The subjects had problems with 10 out of 12. There are 7 logical and 5 illogical CP use cases. The authors and subjects agreed on this. The CP writing rules explicitly consider the logical order and coherence of use cases (CP Style Rules 7 and 8) and this may be a reason for this difference, though group B's Coherence scores (table 5) are slightly lower than group C's, again pointing to differences in problem type. Finally, we note that our 'comprehension test' (phase 4) needs to be developed further and we are preparing another (much larger) experiment to fully explore this important issue.

6 Conclusions

This paper has described a pilot experiment to compare two sets of use case writing guidelines. We tested use cases in terms of efficiency (in time taken to write them and length), application of given guideline rules, and comprehensibility. We found no statistical difference between the guideline sets when comparing efficiency. Both sets performed as well as each other, though we have reservations about this hypothesis (section 5.1). We also found a mixed response to the number of guidelines and rules applied. Overall, there is no significant difference in application, although for CP Structure 1 against CREWS CG5 there is a significant difference for Group A against C ($p=0.03$) and a weakly significant difference for group B against group D ($p=0.05$). When comparing CP Structure 2 against CREWS CG's 1-3, there is no significant difference. CP performs as well as CREWS. In terms of comprehension of use cases, our results reject the hypothesis that CREWS produce more comprehensible use cases than CP. The results indicate that CP does slightly better than CREWS. However, we note that this result was caused by the better performance of group A (ATM), which significantly out-performs group C (ATM) in terms of Coverage and Coherence, and weakly in Consistent Structure. There is no difference between groups B and D (Retail use case). Hence, a more general finding appears to be that problem type has an impact on the performance of the CP Rules.

7 Observations and Further Work

Our attempt to assess the quality of use cases leads us to some general observations. We note that despite the application of guideline sets, all use cases could be seen to have many flaws. This could be due to either weaknesses in the guidelines sets or the use case writers. Indeed, we have altered the CP Rules based upon the response in this experiment (though not for this paper). We also believe that this points to fundamental problems with use cases. By their nature, use cases can be ambiguous, and small errors or flaws can creep in. Use of natural language, though vital to readability and comprehension, exacerbates the ambiguity problem. Furthermore, a lack of structure and semantics means that logic errors seem inevitable. It appears that writing rules provide some guidance to both logic and comprehension. Hence, we believe that further work to examine how to support the writing of use case descriptions is vital. For example, some guidelines (for both sets) appear far more important than others. This suggests the possibility that a minimal set of well used guidelines may be most effective, and this will be a focus for further work. Since, problem type appears to have significant impact we are also considering use case (and scenario) frames. That is, the production of different guidelines sets for different problem types (or frames). Finally, as the logic of use cases seems particularly

vulnerable to human error, we are investigating the use of enactable versions of use cases, which force users to consider the dependencies among events.

We have recently repeated this experiment with a larger sample of 60 subjects. Initial results indicate that the CP Rules have produced similar quality use cases to the CREWS guidelines; that is, there is no statistical difference in terms of comprehension. We have also applied the CP Rules to an industrial case study though we have not compared them against CREWS. However, we have found the CP Rules to be of benefit. We hope to further explore the efficacy of use case guidelines in further case studies.

References

1. Rational Software Corporation: UML Notation version 1.1, http://www.rational.com/uml (1997).
2. Object Management Group: Unified Modeling Language Specification, Proposed Final Version UML 1.3, http://www.omg.org/cgi-bin/doc?ad/99-06-09 (June 1999).
3. Jacobson, I., Christerson, M., Jonsson, P., Oevergaard, G.: Object-Oriented Software Engineering: A Use Case Driven Approach, Addison Wesley (1992).
4. Cockburn, A.: Writing Effective Use Cases, Addison-Wesley (2001).
5. Ben Achour, C.: Guiding Scenario Authoring. Eighth European-Japanese Conference on Information Modelling and Knowledge Bases, Finland, May 25-29 (1998) 181-200.
6. Ben Achour, C.: Writing and Correcting Textual Scenarios For System Design. In: Natural Language and Information Systems Workshop (NLIS'98), Vienna, Austria, 28th August, (1998). http://sunsite.informatik.rwth-aachen.de/CREWS/reports.htm, Report 98-26.
7. Ben Achour, C., Rolland, C, Maiden, N.A.M., Souveyet, C.: Guiding Use Case Authoring: Results from an Empirical Study. In: Fourth IEEE International Symposium on Requirements Engineering (RE'99), University of Limerick, Ireland, 7-11 June, (1999). http://sunsite.informatik.rwth-aachen.de/CREWS/reports.htm, Report 98-31.
8. Weidenhaupt, K., Pohl, K, Jarke, M, Haumer, P.: Scenarios in System Development: Current Practice, IEEE Software, March/April (1998) 34-45.
9. Cox, K., Phalp, K.T.: Replicating the CREWS Use Case Authoring Guidelines Experiment, Int. J. Empirical Software Engineering, Issue 5 (2000) 245-267.
10. Cox, K.: Experimental Material, at the web address: http://dec.bournemouth.ac.uk/staff/kcox/UCwriting.htm, (2001).
11. Gernsbacher, M.A.: The Structure-Building Framework: What It Is, What It Might Also Be and Why. In Britton B.K. and Graesser A.C. (eds.): Models of Understanding Text, Lawrence Erlbaum Associates (1996) 289-311.
12. Graesser, A.C., Swamer, S.S., Baggett W.B.: New Models of Deep Comprehension. In: Britton B.K. and Graesser A.C. (eds.): Models of Understanding Text, Lawrence Erlbaum Associates (1996) 1-32.
13. Graham, I.: Requirements Engineering and Rapid Development, Addison-Wesley (1998).
14. Armour, F., Miller, G.: Advanced Use Case Modelling, Addison-Wesley (2001).
15. Constantine, L., Lockwood, S.: Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design, Addison-Wesley (1999).