CREWS Report 98-31

Fourth IEEE International Symposium on Requirements Engineering
(RE'99), 7-11 June 1999, University of Limerick, Ireland

GUIDING USE CASE AUTHORING : RESULTS OF AN EMPIRICAL STUDY

by

C. Ben Achour*, C. Rolland*, N.A.M. Maiden**, C. Souveyet*

**Centre de Recherche en Informatique*
Université Paris 1 - Sorbonne
90, rue de Tolbiac
75013 Paris - France
{camille,rolland,souveyet}@univ-paris1.fr

***Centre for HCI Design, School of Informatics*
City University
Northampton Square
London EC1V OHB - United Kingdom
n.a.m.maiden@city.ac.uk

Guiding Use Case Authoring : Results of an Empirical Study

C. Ben Achour*, C. Rolland*, N.A.M. Maiden**, C. Souveyet*

**Centre de Recherche en Informatique*
Université Paris 1 - Sorbonne
90, rue de Tolbiac
75013 Paris - France
{camille,rolland,souveyet}@univ-paris1.fr

***Centre for HCI Design, School of Informatics*
City University
Northampton Square
London EC1V OHB - United Kingdom
n.a.m.maiden@city.ac.uk

Abstract

This paper presents results from the first of two empirical studies which examine the effectiveness of guidelines for use case authoring. The ESPRIT 21.903 CREWS long-term research project has developed style and content guidelines for authoring use cases for requirements acquisition and validation. The effectiveness of these guidelines has been evaluated under different conditions. Results indicate that : i. the authoring guidelines improve the overall quality of the use case prose, ii the different guidelines work differently and with different levels of efficiency, and iii use cases are never entirely correctly written ; thus, they can be systematically corrected. The paper details a qualitative and quantitative comparison between guided and non-guided use case authoring. It outlines lessons learned and implications for the CREWS software tools design.

1. Use cases, scenarios and requirements

There has been recently a considerable interest in use cases (UC) and scenarios for acquiring and validating system requirements. [1] recommends describing UCs using narrative text. However, little is known about the advantages and disadvantages of natural language UCs for acquiring requirements. The UML [2] proposes graphical notations such as sequence diagrams to describe UCs, but these notations do not permit expression needed in UCs for effective requirements acquisition and validation [3][4]. This paper evaluates different forms of guidance of natural language expression in scenario-based requirements engineering.

There are clear advantages of narrative text for expressing UCs. Stakeholders are familiar with the notation, and do not need formal training or expensive software tools. However, narrative UCs are often ambiguous and lack structure. Figure 1 presents an example of UC describing distant purchase of goods. The main success scenario is a sequence of action descriptions. This sequence indicates a strict sequence of actions, and the main success scenario is separate from alternative course scenarios and variations [5].

MAIN SUCCESS SCENARIO

1. Buyer calls in with a purchase request.
2. Company captures buyer's name, address, requested goods, etc.
3. Company gives buyer information on goods, prices, delivery dates, etc.
4. Buyer signs for order.
5. Company creates order, ships order to buyer.
6. Company ships invoice to buyer.
7. Buyers pays invoice.

VARIATIONS

1. Buyer may use
   phone in,
   fax in,
   use web order form,
   electronic interchange

7. Buyer may pay by
   cash or money order
   check
   credit card

Figure 1: A simple UC describing the purchase of goods.

There are few guidelines available to make the use of UC more systematic [5][6]. The ESPRIT 21.903 CREWS long-term research project is developing software tools for authoring, and exploiting narrative UCs for eliciting and validating system requirements. It draws on both state of the art practice and current theoretical research in natural language processing. However, using these guidelines introduces a dilemma. Too many prescriptive guidelines will restrict the degree of expression in the UC and make it inadequate for requirements acquisition and validation. On the other hand, too few guidelines will give too much freedom of expression to the author and the quality of UC will decrease, making it difficult for a machine to process. This paper reports on an empirical study of the effectiveness of different types of guidelines to support UC authoring. Results will inform the design of the CREWS-SAVRE [7] and CREWS-*L'Ecritoire* [8] software prototypes.

The remainder of this paper is in 5 sections. Section 2 describes the CREWS approach to UC authoring. Section 3 presents the hypotheses examined in the reported study. Section 4 describes the experimental method. Section 5 reports the results of the evaluation. The paper ends with a discussion of the implications on CREWS tools for UC authoring and analysis.

2. Use Case authoring

2.1 CREWS use case model

CREWS has undertaken an extensive literature review (e.g. [9][5][10][11]) to develop a model of UC, together with a guidance process for authoring UCs [12]. The model is in two parts. The first describes contextual information which links the UC to other UCs, relevant goals, glossaries of terms, and decisions underpinning the design of the UC. The second describes the set of scenarios which constitute the UC. These are divided into a main *normal course scenario* and *extension scenarios* which can be either *normal*, or *exceptional*, depending on whether they allow to reach the associated goal or not.

Every scenario describes a *path of actions* consisting of a sequence of actions. The actions of a path of actions are strictly sequenced and can be constrained by conditions. Scenario actions can be interactions between two agents, or actions of internal type.

CREWS provides guidelines for writing and refining scenarios, and for integrating different scenarios into a UC [12]. Guidelines exist to structure scenarios, make scenarios more complete, precise, and consistent in terminological use [13]. CREWS also provides software assistance to apply the guidelines [8]. It analyses narrative text to 'catch' the semantics of the text and map the text contents into instances of the CREWS UC model. According to Schank [14], analysing the semantics of scenarios can be done, at least partially, through Case Grammars [15]. This view is shared by CREWS who has developed a Case Grammar based on theoretical research in linguistics ([15][14][16][17][18][19]), artificial intelligence [20] and previous applications of Case Grammars to requirements analysis [21]. The linguistic structure and semantics of scenarios actions is interpreted using case patterns and the corresponding set of surface linguistic structures [13].

2.2 CREWS guidelines for use case authoring

To guide in the authoring of scenarios, CREWS provides general guidelines which divide a UC into three sections : i. normal course scenario, describing the normal sequence of actions in the UC, ii. variations to the normal course scenario, and iii. alternative courses to the scenario.

In addition, CREWS provides style and contents guidelines advising on how to write a scenario, and what to put in a scenario. The guidelines are mandatory : they *should* be applied, but may be actually incorrectly applied or not applied at all. However, they are pieces of plain text. In the CREWS-*L'Ecritoire* tool, they can be obtained on demand while writing a scenario. We expect that the quality of the UCs produced improves when the guidelines are correctly applied, but their effectiveness and efficiency needs to be evaluated.

Style guidelines (SG) are derived in part from the CREWS UC model, and in part from current best practice in UC authoring, for example Cockburn's [5] guidelines for structuring UCs. CREWS defines 6 SGs aiming at providing recommendation on the expected form of scenarios :

- **SG1** : write the UC normal course as a list of discrete actions in the form: <action #> <action description>. Each action description should start on a new line. Since each action is atomic, avoid sentences with more than two clauses;
- **SG2** : use the sequential ordering of action descriptions (and hence their unique number identifiers) to indicate strict sequence between actions. CREWS imposes a precise meaning on the ordering of actions in this list. Variations should be written in a separate section;
- **SG3** : iterations and concurrent actions can be expressed in the same section of the UC, whereas alternative actions should be written in a different section;
- **SG4** : use consistent agent, object and action names in all action descriptions in a UC. Avoid use of synonyms and homonyms, and anaphoric references such as he, she, them and it. Be consistent in your use of terminology;
- **SG5** : use present tense and active voice when describing actions;
- **SG6** : avoid use of negations, adverbs, and modal verbs in the description of an action.

Content guidelines (CG) are derived from the afore-mentioned theoretical research in linguistics, artificial intelligence and previous applications of Case Grammars to requirements analysis. The 8 CGs defined by CREWS aim at providing recommendation on the expected content of scenarios :

- **CG1** : <agent> <'move' action><object> from <source> to <destination>;
- **CG2** : <source agent> <'put' action> <object> to <destination agent>;
- **CG3** : <destination agent> <'takes' action> <object> from <source agent>;
- **CG4** : <agent> <action> <agent>;
- **CG5** : <agent> <action> <object>;
- **CG6** : 'If' <alternative assumption> 'then' <action>;
- **CG7** : 'Loop' <repetition condition> 'do' <action>;
- **CG8** : <action 1> 'meanwhile' <action 2>.

Style and content are complementary. A scenario can be written in a good style and its content may be incorrect. The other way round, the content of a scenario can be correct but its style may be inadequate, thus leading to erroneous interpretation. The CREWS hypotheses with regard to the effectiveness and efficiency of the SGs and CGs are thus defined independently.

3. Experimental hypotheses

We propose the following set of hypotheses. The first three hypotheses relate to the use of CGs :

C1 : the use of CGs will lead to UC descriptions which are more correct in terms of the number of complete action descriptions ;

C2 : the use of CGs will lead to less inappropriate action descriptions;

C3 : the use of CGs will lead to UC descriptions which contain more correct and unambiguous descriptions of the flow structure of the UC.

The following four hypotheses relate to the use of SGs :

S1 : the use of SGs will lead to more complete action descriptions ;

S2 : the use of SGs will lead to less unnecessary or inappropriate action descriptions ;

S3 : the use of SGs will lead to UC descriptions which contain more correct and unambiguous descriptions of the flow structure of the UC ;

S4 : the use of SGs will lead to UC descriptions with more consistent use of terminology.

The next section describes how the empirical tasks were conducted and the hypotheses were evaluated using data gathered from the UC authoring exercise.

4. Experimental method

69 software engineers were requested to write a UC each, to describe the interaction between a supermarket checkout system and a checkout operator for purchasing one or more products. Important expected actions in this UC included reading the customer's club card, scanning each product item, requesting the total, and printing an itemised bill. This problem domain was chosen because the UC is of a manageable size and can be written quickly. Furthermore all of the software engineers were familiar with this problem domain through shopping in supermarkets in their everyday lives (checked beforehand using a pre-test questionnaire [22]), thus minimising reliance on the knowledge of the problem domain.

4.1 Experience subjects

The 69 (61 male, 8 female) subjects were full-time or part time post graduate students in Information System engineering at the University of Paris 1 - Sorbonne. In both cases students had received lectures on object oriented modelling, and a half day presentation on UC authoring and modelling. All had professional experience in Information System. Their knowledge extended from object-oriented methods such as UML [2], OBJECTORY [1], OMT [23], O* [24], or Remora [25]. The subjects, aged between 24 and 47 years, volunteered their services and received no financial reward.

4.2 The experimental task

All subjects read a short problem statement that described structure and scope of the supermarket checkout problem domain. Subjects were asked to write a complete UC for interaction with the supermarket checkout system, including normal and alternative courses. The set of 69 subjects was divided into 4 groups as shown in Table 1.

Table 1 : The four experimental groups.

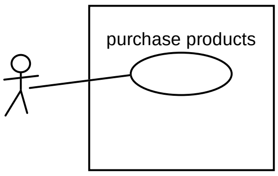
Group	Definition
A	a control group in which subjects were given the problem statement describing the supermarket checkout problem domain only
B	an experimental group in which subjects were given the problem statement and CREWS SGs
C	an experimental group in which subjects were given the problem statement and CREWS CGs
D	an experimental group in which subjects were given the problem statement and both CREWS SGs and CGs

Subjects were balanced across groups according to their gender (M/F), to their experience in systems analysis/design (which ranged from 17 years to no professional experience) and to their experience in programming (which ranged from 16 years of professional experience to no previous programming experience). Equal numbers of subjects with and without this experience were balanced across groups.

Subjects were given 60 minutes to write the UC as a pilot study indicated that this was sufficient time to complete the task. Prior to undertaking the task, experimental instructions [22] were read by the experimenter. Then, subjects were then permitted to read the problem statement shown in Figure 2. Each subject was instructed to produce a UC, and subjects of groups B, C, and D were informed that the guidelines provided to them should be used. However, details about the way in which guidelines should be used were not given. All subjects were informed beforehand of the time limit on the task and were permitted to seek clarifications during the task.

Subjects were given sheets of paper on which to write their names and UCs. The answer sheets had no predefined structure which might have influenced the structure or content of their UC.

A supplier of supermarket checkout machines is to produce a new checkout machine. It wants to use UML notation to develop the software system in the machine. A requirements analysis has revealed a number of important functions. A checkout operator must use each checkout machine to record purchases and receive payments from customers (including the use of club cards, payment using credit cards and the printing of receipts). An operator must also be able to use the checkout to call the store manager when problems arise. The use case model for the system including all actors and use cases is shown.



Please produce a use case description for the use case for handling customer purchases. Show the basic course and all relevant variations and alternative courses. Please remember there is no right answer, so make all relevant assumptions clear.

Figure 2 : Problem statement.

4.3 Evaluation scheme

All the subjects' UCs were evaluated against different criteria so as to refute or confirm each of the 7 experimental hypotheses presented in section 3.

Hypotheses S1 and C1 : action descriptions in the subject's UC were scored for completeness with respect to the most appropriate action statement structure indicated in CG1 to CG8. A score was also allocated for each predicted element found missing in the action descriptions. Missing elements were expected to be agents, objects, and communication sources or destinations. For instance, the action :

'The product price is communicated to the system'

was counted as incomplete, as it does not indicate who or what is source of the communicated *'product price'*.

Hypotheses S2 and C2 : completeness scores were allocated to each subject's UC. To construct a marking scheme, an 'expert' UC was developed by the authors using their more in-depth knowledge of supermarket checkout systems [22]. The marking scheme contained a list of relevant actions to be included in the UCs, and focused on the semantics of each action rather than the syntax and grammar. Subjects' scores were augmented for each action in their UC that was also in the expert UC. For example, the following extract of a subject's UC :

'the operator presents the bar code of the product to the optical reader'

was identified as a relevant action, and the corresponding UC completeness score incremented of one. Indeed, this action corresponds roughly in the expert UC to the action :

'the operator swipes the product label using the bar code reader'.

Although expressed very differently, the two actions identify the same real world event : the bar code label of a product is put in front of an optical bar code reader. Both actions involve the same agents : the *'operator'* and the *'optical reader'* of the checkout system (named *'bar code reader'* in the expert UC), and the same objects : the *'bar code of the product'* (referred as the *'product label'* in the expert UC).

It happened that some actions which were not identified in the expert UC were however counted as relevant in the completeness score. This was the case for instance in a surprisingly innovative UC describing a checkout system able to count change and give it back directly to the customer. Although not initially foreseen in the expert UC, the action describing the automatic change return was counted as relevant. Indeed, it does participate to the description of the supermarket checkout system behaviour. Therefore, the subject's UC completeness score was incremented.

Hypotheses S3 and C3 : the marking scheme also contained a list of keywords to indicate flows of actions (i.e. sequenced, constrained, and iterative actions) which were predicted in the expert UC. Subjects' score was augmented for each predicted keyword that was included correctly in their UC.

Furthermore, each subject also received a score indicating the number of variations incorrectly positioned in the UC with respect to the logical/required position. A frequent example of incorrectly written variation was for instance the statement of the cases :

'if the customer pays by cash ...', and
'if the customer pays by check ...', and
'if the customer pays by card ...'

in the same scenario. The three kinds of payments, being exclusive, were expected to be considered as variations. They were thus expected to be described in different scenarios, as alternative flows of actions.

So as to complete evaluation of hypothesis S3, irrelevant variations (i.e. variations that were not expected in the 'expert' UC) were scored for each subject.

Hypothesis S4 : was measured by the number of synonyms and homonyms present in the UC. Each subject received a score for each occurrence of a synonym pair and a homonym pair in the UC.

A *homonym* is defined as one term which is used to describe two different objects. This happens in UCs that use the same name for both a real-world object (e.g. the physical product) and a model of that object in the software system (e.g. the information about that product in the checkout system). Another example of homonym is the word *'change'* in the actions :

'the system displays the change to the operator', and
'the operator gives the change back to the customer'

On the contrary, a pair of *synonyms* corresponds to two different terms used to identify a single object. For example, *'customer'* and *'client'* are synonyms. In the subject's UCs, the *'system'* was also referred to as the *'checkout system'* (the difference might not seem obvious in English, but the French words are very different : *'système'* and *'caisse enregistreuse'*).

S4 was also evaluated using the number of occurrences of pronouns such as *'he'*, *'she'*, *'it'*, *'his'*, *'him'* in all action descriptions of subjects UCs.

All subjects completed the task and answered the retrospective questionnaire about their UCs specifications and reuse behaviour. All the UCs were analysed according to the evaluation criteria mentioned above. So as to ensure quality and coherence of the results, all UCs were analysed at least twice by different correctors. The results are presented in the next section.

5. Evaluation

The measurement work of the evaluation criteria for the 69 UCs resulted in a table containing 1341 quantitative and qualitative data. The statistical analysis of these results was twofold : (a) qualitative search of homogenous populations among subjects, and (b) quantitative measurement of differences between group A and the other groups, based on T-tests.

T-tests explore whether the probability of two results arising from different populations is less than a significant level (e.g. 5%) or not [26]. Beforehand, the normality of evaluation criteria was checked for each group (this is a pre-condition for getting relevant results from T-tests). Although quite low, the group size was still relevant to apply the test ; all variables were following a normal law. Key results from the tests are shown in Table 2. The table shows some significant validation of the hypotheses :

- *S1/C1* : subjects of groups C and D wrote UCs with significantly fewer missing elements in actions than subjects from group A (0.01 for both).
- *S2/C2* : no significant improvement of UC completeness for subjects of groups B, C and D was found.
- *S3/C3* : subjects of group B wrote UCs with significantly fewer variations incorrectly positioned within the normal course than those of group A
- *S4* : for anaphors, group C used fewer anaphors than group A nearly significantly (0.09).

Table 2 : Key impact of the guidelines on UC evaluation criteria (++ = significant improvement).

Evaluation criteria	Groups B and D	Groups C and D
Action completeness		++
UC completeness		
UC flow structure correctness	++	
Terminology consistency		++ (especially anaphors)

These primary results show that some of the hypotheses are verified. The guidelines do help increase the completeness of UCs, and stop you making errors when writing UCs. However, the quantitative analysis does not seem to indicate that the guidelines worked significantly in all cases. The rest of this section presents a detailed analysis of the results of the experiment.

5.1 Detailed evaluation of hypotheses S1/C1

Because they are implicit in the text, missing elements of incomplete actions often raise the problem of ambiguity. To give a scale of size of the frequency of the error :

- 41% of subjects have written less than 3 complete actions out of 10 in their UCs,
- 49% of subjects have written from 4 to 6 complete actions out of 10 in their UCs, and
- 10% of subjects have written more than 7 complete actions out of 10 in their UCs.
- Only two subjects have not written a single incomplete action in their UCs.

Linguistically, 57% of the elements that were missing in incomplete action descriptions were agents ; 38% of missing elements were either the source or destination of actions describing communications. Conceptually, 85% of UC actions that were not completely described were actually missing one of their two agents.

Figure 3 describes the distribution of subjects having more than 1 complete action out of 10 in their UC, more than 2 out of 10 complete actions, and so on for each group. The figure shows better results for groups B, C and D than for group A.

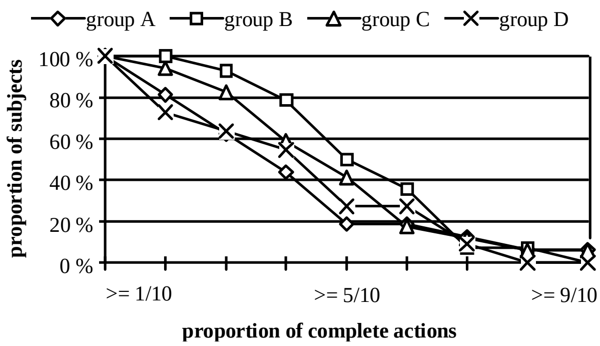


Figure 3 : Distribution of subjects in function of the proportion of complete actions per group.

Figure 3 shows that *hypothesis S1 is, at least partially, verified* : whereas only 20% of subjects from group A had more than half of their actions complete, 50% subjects of group B have correctly completed half of the actions in their UCs. This result is not in contradiction with those from Table 2 : in fact, subjects of group B wrote incomplete actions less frequently, but their incomplete actions miss in average more elements.

Concerning hypothesis C1, the results of Figure 4 enforce those of Table 2 : 35% of subjects from group C had more than half of their actions complete.

5.2 Detailed evaluation of hypotheses S2/C2

UC completeness is important : the more complete a UC is, the more requirements it helps to identify. According to the UC completeness criteria, hypotheses S2 and C2 are verified (see Table 2). However, many actions in subjects' UCs were not predicted in the expert UC. Either these actions were outside the scope of the supermarket checkout system problem domain, or they belonged to an inadequate level of abstraction (too abstract/too detailed). Examples of unpredicted actions were :

i. *'the operator puts the product in a bag'*, and

ii. *'the operator cancels the transaction'*.

The former was considered as irrelevant because it is external to the checkout system. The latter action only implicitly involve the checkout system : to cancel the transaction, the operator has to act on the system, e.g. by pressing a 'cancel transaction' button. This action was thus counted as being not expressed at the right level of abstraction.

In the framework of real world requirements engineering, such actions would have incorrectly inflected the scope of the activity, and possibly driven to incorrect design. Therefore, each subject received also a score for unpredicted actions. These were categorised according to those irrelevant or those not at the expected level of abstraction.

It is noteworthy that the dividing line between irrelevant actions and those at inappropriate abstraction levels is very thin. For example, action (i) above is irrelevant. However, consider (i) rephrased as follows :

i.' *'the operator puts the product on the conveyor belt and the checkout system conducts the product into the bag'*

Now the checkout system supports explicitly action (i'). Used in the sense of (i'), the action (i) would be considered, not as an irrelevant action, but as an action described at an incorrect level of abstraction.

Figure 4 aims at refining the evaluation of hypotheses S2/C2 already proposed in Table 2. In so far as the level of abstraction of UC actions is concerned, Figure 4 confirms the results of Table 2. Indeed, SGs and CGs both improve the proportion of relevant actions, but not in a significant way. Regarding action relevance, only results of group B, are significantly better than those of group A (the improvement is of 13%).

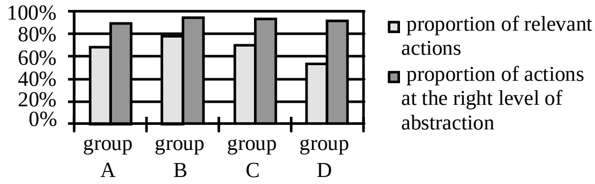


Figure 4 : Average proportion of relevant actions and of actions at the right level of abstraction per group.

The results of the evaluation of hypotheses S2 and C2 are thus rather mixed. Whereas the guidelines do not significantly improve UC completeness, SGs do diminish the proportion of actions that are out of the scope of the problem domain.

5.3 Detailed evaluation of hypotheses S3/C3

For about 45% of subjects' UCs, more than half of the explicit flows of actions were incorrectly written. Figure 6 describes the distribution of subjects who correctly wrote more than 1 correct flow out of 10 in their UC, more than 2 out of 10, more than 3 out of 10, and so on for each group. The figure does not show better results for subjects of group B than for subjects of group A, except that two times more subjects of group B had all their flows correctly written. On the contrary, the results of group C are significantly improved with respect to group A (80% of UCs by group B had half flows correctly written), and results of group D even better than those of group C (50% of subjects from group D had all their flows of actions correctly written).

The foregoing results mean that CGs do improve the structure of authored scenarios. As such, SGs do not improve scenario structure, but act as a good explanatory complement to CGs.

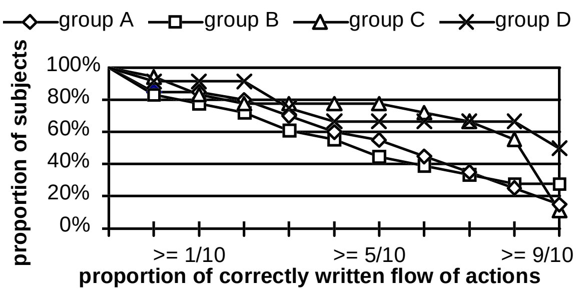


Figure 5 : Distribution of subjects in function of the proportion of correctly written flow of actions.

Figure 5 deals with the flow structure of the UC within scenarios. So as to cover all the aspects of the UC flow structure, we must also consider the aspect of the organisation of the various scenarios in the UC. Figure 6 presents for each group the proportion of subjects having no incorrectly written variation in their UCs. The correctness of variations was, as mentioned earlier, evaluated with respect to their position in the UC and to their relevance. The table shows a significant improvement for groups B and D in comparison with group A.

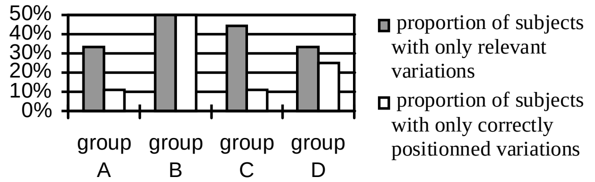


Figure 6 : Proportion of subjects having written correctly all UC variations per group

The validity of hypothesis S3 is reinforced (with respect to Table 2). Additionally, hypothesis C3 is validated. The SGs lead to UC descriptions in which variations are more correctly described ; the flow structure of the UC is thus improved. However, the flow structure of individual scenarios within the UC is not improved by SGs themselves except when those are used to complement CGs.

5.4 Detailed evaluation of hypothesis S4

Experience seems to tell that in writing requirements there are many words and phrases that introduce ambiguity and make the requirements vague and unverifiable. We have identified that 50% of UCs were containing terminology errors : synonyms, homonyms, and ambiguous use of pronouns. The most frequent one is the use of pronouns (74% of terminology errors), followed by synonyms (21%). Table 2 shows that SGs diminish significantly the rate of anaphoric references in UCs. According to Figure 7, subjects of group B have used anaphoric references 60% less frequently than subjects of group A. In so far as synonyms are concerned, we identified a slight improvement only. Homonyms were rare in UCs, and consequently no significant difference between the groups could be identified.

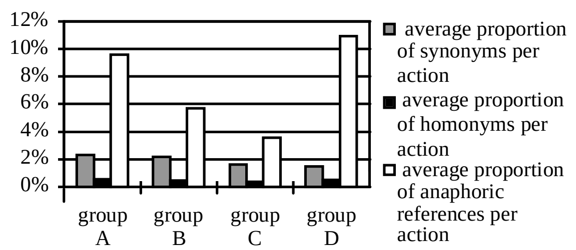


Figure 7 : Proportion of terminology errors per action for subjects of each group.

Hypothesis S4 is thus confirmed. Figure 6 also shows improvements in terminology use due to CGs. However, a combination of SGs and CGs does not always provide better results. Discussion with the subjects after the experiment showed that subjects of group D had difficulty in managing all the guidelines at the same time. Evidently, SGs and CGs are independently efficient, but difficult to apply in conjunction

6. Discussion

The key results are summed up in Table 3. The table shows that : (i.) guidelines do improve UC authoring, (ii.) all guidelines do not improve all UC features.

Table 3 : Key results of the evaluation

Improved UC feature	Do SGs help ?	Do CGs help ?
Action completeness	(S1) yes, proportionally to the number of actions	(C1) yes, significantly, but combined, SGs and CGs are less efficient than used separately
UC completeness	(S2) the proportions of relevant actions and of actions at the right level of abstraction are significantly improved	(C2) results similar to S2
UC structure	(S3) significantly for the structuring of UCs into main course scenario/variations	(C3) significantly for the internal flow structure of scenarios in a UC
Terminology correction	(S4) slightly for synonyms and significantly for anaphoric references	yes, but not as supplement to SGs.

Regarding (ii.) : since improvements due to guidelines are always specific to some UC features, combining guidelines is necessary for all round improvement of UCs. However, the experiment showed also that too many guidelines are difficult to manage, and actually diminish their specific effectiveness. The combination of guidelines should thus be transparent to UC authors.

Regarding (i.) : whereas they are effective for authoring better UCs, guidelines very seldom lead to perfect UCs. Checking UCs is therefore necessary whenever their quality is important.

In conclusion, the experience showed that the guidelines are usable, applicable, relevant and useful, though non guided UC are more inventive. All this was conjecture before the study. Two key issues emerged from the experience : the training of UC authors, and the presentation of guidelines to maximise adherence to them. The impact on the CREWS prototype is twofold.

First, guidelines should be implemented so as to appear more transparent to users. In its current form, the prototype displays guidelines on request as editable texts. Other design options could be : i. to implement CGs as templates to be filled up, ii. to provide guidelines to UC author only when necessary, e.g. as tips displayed when an error is detected, or iii. to use guidelines in a more dynamic way, e.g. by walking through scenarios using CGs to check errors, or by complementing SGs and CGs with dynamic guidelines like 'scenarios for authoring UCs'.

Second, discussion with the subjects after the experiment showed that they were not always able to interpret the guidelines. Some subjects asked us to confirm their understanding of the guidelines during the experiment. The wording of guidelines is important, but we also believe that the guidelines should be clarified with explanations. For example, counter examples emphasising errors to be avoided would be useful. Similarly, justification of why guidelines should be respected, and the impact on design of not respecting a guideline could make UC authors more careful. However, to stay efficient, guidelines need to remain as short as possible. A solution to the problem would for instance be to use a hypertextual presentation.

Acknowledgements : we would like to thank the students from the DESS SI of the IAE for their dynamic contribution to this experiment.

References

[1] I. Jacobson, *The Use Case Construct in Object-Oriented Software Engineering*, In John M. Carroll (ed.), *'Scenario-Based Design: Envisioning Work and Technology in System Development'*, John Wiley and Sons, pp 309-336. 1995.

[2] M. Fowler, K. Scott. *UML Distilled, Applying the Standard Object Modeling Language*. Addison Wesley, 1997.

[3] C. Potts, K. Takahashi and A.I. Anton, *Inquiry-based Requirements Analysis*. in IEEE Software 11(2) (1994) 21-32.

[4] M. Kyng, *Creating Contexts for Design*. In John M. Carroll (ed.), *'Scenario-Based Design: Envisioning Work and Technology in System Development'*, John Wiley and Sons, pp. 85-107, 1995.

[5] A. Cockburn, *Structuring Use Cases with Goals*. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1, http://members.aol.com/acockburn/ papers/usecases.htm, 1995.

[6] I. Graham, *'Migrating to object technology'*. Addison Wesley, 1995.

[7] N.A.M. Maiden, *SAVRE : Scenarios for Acquiring and Validating Requirements*. Journal of Automated Software Engineering 5, pp.419-446, 1998.

[8] C. Rolland, C. Souveyet, C. Ben Achour, *Guiding Goal Modelling using Scenarios*. Accepted for IEEE Transactions on Software Engineering, Special Issue on Scenario Management, 1998

[9] J.C.S. do Prado Leite, G. Rossi, F. Balaguer, A. Maiorana, G. Kaplan, G. Hadad and A. Oliveros, Enhancing a requirements baseline with scenarios, In Third IEEE International Symposium On Requirements Engineering (RE'97), Antapolis, Maryland (IEEE Computer Society Press, 1997) 44-53.

[10] B. Dano, H. Briand, F. Barbier, *A Use Case Driven Requirements Engineering Process.* Proc. Third IEEE International Symposium On Requirements Engineering (RE'97), Antapolis, Maryland (IEEE Computer Society Press), 1997.

[11] J. Rumbaugh, G. Booch, *'Unified Method, Notation Summary'* Version 0.8, Rational Software Corporation, 1996.

[12] C. Rolland, C. Ben Achour. *Guiding the Construction of Textual Use Case Specifications*. In Data & Knowledge Engineering Journal, Vol 25, N°1-2, (ed. P. Chen, R.P. van de Riet), North Holland, Elsevier Science Publishers, pp. 125-160, March 1998

[13] *C. Ben Achour. Guiding Scenario Authoring.* Proc. 8th European- Japanese Conference on Information Modelling and Knowledge Bases, Finland, (ed. H. Jaakola, H. Kangassalo), pp. 181-200. Mai 1998.

[14] R.C. Schanck, *Identification of conceptualisations underlying natural language*. In *'Computer models of thought and language'*, R.C. Shanck, K.M. Colby (eds), Freeman, San Francisco, pp. 187-247, 1973.

[15] C. Fillmore, *The case for case.* In E.Bach, R. Harms (eds.) 'Universals in linguistic theory', Holt, Rinehart and Winston Publishing Company, pp. 1-90, 1968.

[16] R. Simmons, *Semantic Networks : their Computation and Use for Understanding English Sentences*. In *'Computer Models of Thought and Language'*, R.C. Schanck, K.M. Colby (eds), Freeman, San Francisco, pp. 63-113, 1973.

[17] Y. Wilks, *Good and Bad Arguments about Semantic Primitives*. Report n° 42, Department of Artificial Intelligence, University of Edinburgh, 1977.

[18] B. Boguraev and K. Spark-Jones, *A Note on a Study of Cases*. In *Computational Linguistics*, Vol 13, n° 1-2, pp. 65-68, 1987.

[19] S.C. Dik, *'The Theory of Functional Grammar, part I : the Structure of the Clause'*. Functional Grammar Series, Fories Publications, 1989.

[20] J.F. Sowa, *'Conceptual Structures, Information Processing in Mind and Machine'*. Addison-Wesley publishing company. 1984.

[21] M. Saeki, H. Horai, H. Enomoto, *Software Development Process from Natural Language Specification*. Proceedings of the 11th International Conference on Software Engineering, pp. 64-73, 1989.

[22] C. Ben Achour, N.A.M. Maiden. *Empirical Study of Use Case Authoring : Experimental Material*. Internal Report, Centre de Recherche en Informatique, Université de Paris 1 - Sorbonne, http://www.univ-paris1.fr/CRINFO/users/benachour/ ESEM/index.html, 1999.

[23] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *'Object-oriented Modelling and Design'*. Prentice Hall, 1991.

[24] J. Brunet, *Analyse Conceptuelle Orientée Objet*. Phd thesis, University of Paris 6 - Pierre and Marie Curie, 1993.

[25] C. Rolland, O.Foucaut, G. Benci, *Conception des Systèmes d'Information : la Méthode REMORA*. Eyrolles, 1988.

[26] B. Escofier, J. Pagès, *Initiation aux Traitements Statistiques, Méthodes, Méthodologies*. Presses Universitaires de Rennes, 1997.