Human-Computer Interaction -- INTERACT'03
M. Rauterberg et al. (Eds.)
Published by IOS Press, (c) IFIP, 2003, pp. 892-895

Evaluation of an Advisor Tool for Scenario Generation

Jae Eun Shin, Alistair Sutcliffe & Andreas Gregoriades

Centre for HCI Design, Department of Computation, UMIST, Sackville Street
Manchester, M60 1QD, UK

jeshin@co.umist.ac.uk

Abstract: This study investigates the usefulness of a scenario advisor tool which was designed to be used in the military domain where human or machine errors cause safety-critical problems. The tool provides traceability between scenario models and requirements and helps to generate new scenarios and scenario variations. Through two series of evaluation sessions, we found that the tool is useful to generate scenarios.

Keywords: scenario generation, user evaluation, scenario schema, scenario taxonomy

1 Introduction

The scenario-based approaches to design have become popular for eliciting and validating requirements (Carroll, 2000,1995; Rolland et al., 1998; Sutliffe et al., 1998; Weudebgayot et. al., 1998). However there are few methods or tools to guide scenario based design; furthermore eliciting or generating a sufficient set of scenarios can be difficult.

Cunning and Rozenblit (1999) stated that “incomplete, ambiguous, incompatible and incomprehensible requirements lead to poor designs,” and developed a method for generating test scenarios from the structured requirements specification. Use of sound scenarios is crucial for requirements elicitation and validation, but their method generates many possible event scenarios, as a tree structure, based only on the specification and supplied constraints. There seem to be no rigorous methods or tools to generate sufficient sets of scenarios and this motivated the current study to develop an advisor tool for scenario generation.

The scenario advisor tool consists of a scenario annotation editor and a hypertext scenario schema for traceability and scenario generation help. The annotation editor allows users to annotate scenario narratives from scenario schema based on the ***i**** model (Mylopoulos, 1998; Yu, 1997) and derived from relevant literature (Carroll, 1995; Carroll et al., 1994; Daren, Harrison & Wright, 2000; Mylopoulos, 1998; Sutcliffe et al., 1998). This paper focuses on experiments into the tool’s functions related to scenario generation.

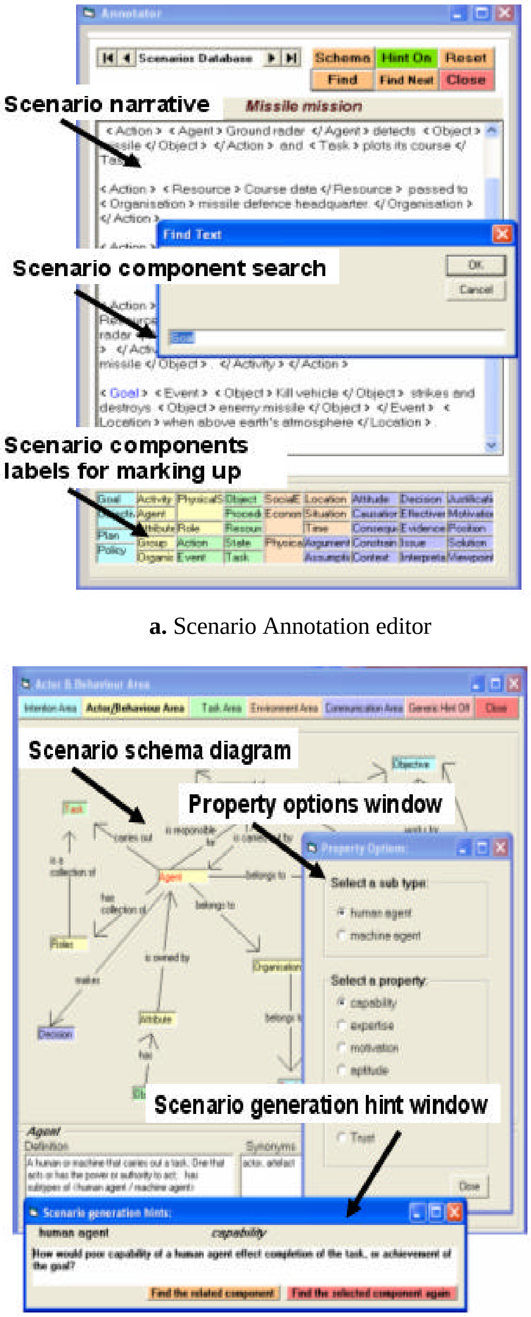
2 Scenario Advisor Tool

The tool has two main features: (1) traceability between scenario narratives and model components, and (2) help for scenario generation.

Traceability increases trust in the model by capturing traces between requirements and model. Without traceability information, the usefulness of models is severly limited (Egyed, 2001). To operate the tool for traceability, we select a scenario narrative marked up with scenario components (Figure 1a). Double-clicking any component tags (e.g. <task>, </task>) from the scenario narrative, the tool opens a relevant schema diagram and highlights the related components.

More importantly, the advisor tool helps users to generate new scenarios or produce variations on existing scenarios by providing scenario generation hint questions (e.g. How does reliability of machine agent affect achieving a task?). To check the scenario generation hint questions, we click on a certain component node (e.g. Physical Environment) from a schema diagram, then the tool shows a pop-up window listing properties of the selected node (e.g.

whether state, climate, noise, interruption, fatigue, stress). By selecting one of the properties, the tool shows scenario generation hint questions with an example (Figure 1b). The answer to the scenario generation hint questions is an event of a scenario, so sets of answers to the hint questions will form a new scenario or a scenario variation.



< Action > < Agent > Ground radar </ Agent > detects < Object > missile </ Object > </ Action > and < Task > plots its course </ Task >

< Action > < Resource > Course data </ Resource > passed to < Organisation > missile defence headquarter. </ Organisation > </ Action >

< Goal > < Event > < Object > Kill vehicle </ Object > strikes and destroys < Object > enemy missile </ Object > </ Event > < Location > when above earth's atmosphere </ Location >

Agent
Definition
A human or machine that carries out a task. One that acts or has the power or authority to act; has subtypes of (human agent / machine agent)

human agent capability
How would poor capability of a human agent effect completion of the task, or achievement of the goal?

a. Scenario Annotation editor

b Scenario schema

Figure 1: Screen dumps of the advisor tool

3 Evaluation Method

There were twenty individual evaluation sessions. Ten participants were asked to complete a set of tasks individually without the scenario advisor tool, and the remaining ten individuals with the tool.

The tasks used for both experiments with and without the tool were identical. The experimental task consisted of two sub-tasks. First, the participants were asked to write new scenarios based on the same 'Missile Mission' scenario provided with both plain and marked-up narratives. They were then asked to generate scenario variations of the 'Missile Mission' scenario.

All the sessions started by participants completing a pre-test questionnaire to collect user profiles. The users were then asked to read the instruction handouts. The experimenter briefly demonstrated the tool or paper-based information for 5 minutes and users had 10-minutes familiarisation time. During the familiarisation time, the users were asked to complete a training task for finding scenario components and properties from the information provided. They were then asked to complete the experimental task (writing new scenarios and scenario variations) within 30 minutes. All the sessions were audio-recorded while one experimenter observed the participants completing their tasks. After completing the task, each participant was asked to fill out the post-test questionnaire. The individual sessions ended with debriefing interviews.

3.1 Generating Scenarios without the Tool

Eight postgraduates and two researchers (mean age = 28 years; mean computer use = 10.3 years; 6 male / 4 female) participated in the experiment without the tool. None of them had experience in scenario-based design, although three had some experience in writing scenarios at novice level. Only two participants said they were familiar with the military domain.

Individual participants were provided with paper-based information (scenario taxonomy table and scenario schema diagrams) instead of the scenario advisor tool we built. The participants were asked to use the paper-based information while they were completing the tasks.

3.2 Generating Scenarios with the Tool

Nine postgraduates and a researcher (mean age = 27.5 years; mean computer use = 10.5 years; 6 male / 4 female) participated in the experiment without the tool. None of them had experience in scenario-based design, although five had some experience in writing scenarios at novice level. Three participants said they were familiar with the military domain, but there was no domain expert.

The scenario advisor was run in a laptop (Windows 2000; normal mouse operating) in the laboratory. Participants were asked to use the tool to get advice while they were completing the task.

4 Results and Analysis

We used the observation notes and the audio-recordings of evaluation sessions to find out users' scenario generation strategies and any usability problems. Questionnaires and debriefing interviews were for follow-up problems, suggestions and design ideas.

4.1 Task Performance

All twenty participants completed the experimental tasks although two could not finish the whole tasks successfully within the time given.

The user answer sheets were assessed using pre-defined standard solutions. For each sub-task, we expected users to reach a maximum 50 points. The answers were marked as follows:

- 10 points for using scenario components different from the original scenario
- 10 points for using scenario schema areas different from the original scenario
- 10 points for the length of the scenario
- 10 points for different themes or plots
- 10 points for a good and detailed story

Participants were paired in teams 1 to 10 for the purpose of comparing task performance. Each team includes one user without the tool and the other with the tool.

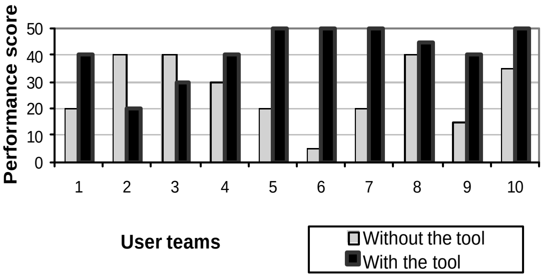


Figure 1: Task Performance (Generating new scenarios)

As shown in Figure 1, user performance for Task A (Writing new scenarios) were mostly higher with the tool, with the exception of two users (Users 2 and 3). This is because users with the tool could find any information about the scenario components and their relationship more easily with the hypertext tool, and the traceability function helped users to find the relevant information more effectively. User 2 showed lack of motivation completing the task because she did not want to think or write about war. Overall, user performance was found to be highly significant from the t-test (P value: 0.008).

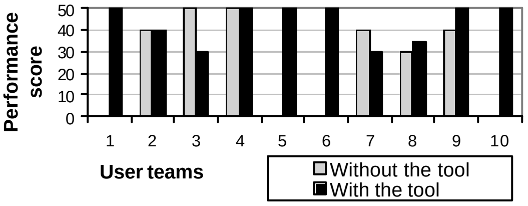


Figure 2: Task Performance (Generating scenario variations)

User performance for Task B (Writing new scenarios) was generally better with the tool, except for three users, as shown in Figure 2. It was found that the providence of scenario generation hint questions improved the performance of with-the-tool users. The main reason for lower performance for Users 3, 7, and 8 (with the tool) and Users 1, 5, 6, and 10 (without the tool) was lack of domain knowledge. For this task performance, the t-test shows the two means (with and without the tool performance) significantly different (P=0.031).

4.2 Usability Problems

The usability problems found from the evaluation were categorised into four subsets: software bugs, interface problems, content problems, conceptual problems and missing requirements. As software bugs, users found a typological error and one error showing properties for agent in the Environmental area. The schema window closed unexpectedly every time when closing the 'Hint' window. For interface problems, two users could not find it easy to reveal the property option window from the schema. The other problems found are:

Content problems

- Incorrect property list for machine agent
- Unfamiliar terminology like schema, component area, related components

Misleading cues

- Confusion in finding same components in different areas from schema diagrams (e.g. <agent> component in [Actor& Behaviour] or [Task] area
- Confusion about functions of component labels used for tagging (e.g. a user tried to use this button to reveal relevant area)

Missing requirements

- Need scrollable or resizable textboxes for definitions, synonyms and properties in the schema window
- Need help for software usage
- Need to automatically reset highlighted mark-up tags from the annotation editor for new component search

Despite the above problems, users said they found the tool very useful in general and their satisfaction level shown from post-test questionnaire was high (Team 2, 4 & 8 between 4.1 and 4.6; The remaining six teams between 6.4 and 6.6).

5 Conclusion

This study demonstrated that the scenario advisor tool helped users to write sound scenarios without any domain knowledge, and to generate more reliable variations on existing scenarios by providing scenario generation hints for each property of model components. Simple hint questions appears to be sufficient help for developing scenario variations and this enable more productive validation of interactive systems.

We also found from debriefing interviews that the most difficult parts of the evaluation sessions for all users were the lack of domain knowledge and little experience in writing scenarios. Therefore, future work should concentrate on developing help systems for step-by-step scenario generation procedure and domain-specific information.

Acknowledgement

This work was funded by EPSRC Systems Integration Programme SIMP project (Systems Integration for Major Projects).

References

Carroll, J.M. (2000). *Making Use: Scenario-based Design of Human-computer Interactions.* MIT Press, Cambridge, MA.

Carroll, J.M.(1995), *Scenario-based Design: Envisioning Work and Technology in System Development.* John Wiley, New York.

Carroll, J.M., Mack, R.L., Robertson, S.P, and Rosson, M.B. (1994). Binding Objects to Scenarios of Use. *International Journal of Human-Computer Studies,* 41, pp.243-276.

Cunning, S.J. and Rozenblit, J.W. (1999). Test Scenario Generation from a Structured Requirements Specification, Proceedings of IEEE Conference and Workshop on Engineering of Computer-Based Systems. March 7 -12, Nashville, Tennessee. pp. 166-172

Daren, A, Harrison, M. and Wright, R. (2000). Allocation of Function: Scenarios, Context and the Economics of Effort. *International Journal of Human-Computer Studies,* 52, pp.289-318.

Egyed, A. (2001) A Scenario-driven Approach to Traceability. *Proceedings of the 23rd International Conference on Software Engineering (ICSE),* Toronto, Canada, May 2001, pp.123-132

Mylopoulos, J. (1998). Information Modelling in the Time of the Revolution. *Information Systems* 23, pp.127-155.

Rolland, C. Arhur, B.C., Cauvel, C., Ralyte, J., Sutcliffe, A.G., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E., & Heymans, P. (1998). A Proposal for a Scenario Classification Framework. *Requirements Engineering,* 3, pp.23-47.

Sutcliffe, A.G., Maiden, N.A.M., Minocha, S., & Manuel, D. (1998), Supporting scenario-based requirements engineering. *IEEE Transactions on Software Engineering*, 24(12), pp.1072-1088

Weidenhaupt, K., Pohl, K., Jarke, M., & Haumer, P. (1998) Scenarios in System Development: Current Practice. IEEE Software, March/April, pp.34-45

Yu, E. (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In *Proceedings of the 3rd IEEE Int. Symposium on Requirements Engineering.* IEEE Computer Society Press, Los Alamitos, CA., pp.226-235