Setting a Foundation for Collaborative Scenario Elicitation

Ann M. Hickey
ahickey@cmi.arizona.edu
Douglas L. Dean
ddean@cmi.arizona.edu
Jay F. Nunamaker, Jr.
nunamaker@bpa.arizona.edu
Center for the Management of Information, University of Arizona, Tucson, AZ 85721

Abstract

Eliciting requirements from users remains a major challenge for systems developers. The Collaborative Software Engineering Methodology addresses this challenge by combining the use of scenarios with Group Support Systems (GSS) technologies. This paper reports results of an experiment conducted to analyze definition of those scenarios using a general-purpose GSS. Scenario quality and productivity were evaluated and then compared for three textual scenario formats. Results highlighted the productivity of the unstructured format as well as the scenario completeness problems of all formats. Recommendations for an iterative collaborative scenario process and a special-purpose GSS scenario tool were developed to address these problems.

1. Introduction

Several decades of MIS research have clearly shown both the importance and the difficulty of developing complete and accurate information systems requirements. Many of the seemingly endless reports of spectacular system failures have been attributed to requirements problems [1]. Other studies have shown that problems are 200 times more expensive to correct during testing than during the requirements phase [2]. Most researchers and practitioners agree that user involvement is critical to the success of the requirements process. The challenge is in determining how to effectively involve multiple users in that process.

Traditional requirements elicitation techniques depend on user interviews or group meetings as the primary mechanisms for involving users. However, these techniques are notoriously inefficient, especially when dealing with large, highly diverse user groups. The problem of gathering accurate requirements and the inefficiencies of user interviews and group meetings were some of the early driving forces in Group Support Systems (GSS) research. GSS have been highly successful in improving group meeting productivity and outcomes in real-world settings [3]. On-going research on the use of GSS for collaborative requirements elicitation has led to the development of special-purpose GSS tools for activity and data modeling and an over-arching Collaborative Software Engineering Methodology (CSEM) [4]. However, development of CSEM identified a need for additional collaborative tools to support key aspects of the requirements elicitation process. The Activity Modeler provides a structured mechanism for establishing the scope of the system and identifying the functions that it should support [5]. The Group Data Modeler has proven effective for capturing data requirements [6]. Neither provides a dynamic picture of the business processes that a system must support, i.e., the system's behavioral requirements. Recent research has suggested that scenarios may be an effective mechanism for eliciting behavioral requirements. However, although there is widespread agreement on the usefulness of scenarios, there are many unanswered questions on how to integrate scenarios into a collaborative requirements elicitation process such as CSEM. The purpose of the experiment reported in this paper was to set a foundation for collaborative scenario elicitation by exploring the first step in that process, individual user definition of scenarios. Future research will focus on the collaborative scenario review and evolution needed to fully incorporate scenarios into CSEM.

During the experiment, subjects individually defined scenarios using a general-purpose GSS, GroupSystems Group Outliner. Overall scenario quality and user productivity were evaluated and then compared for three different textual scenario formats. The experimental results were further analyzed to assess the implications for a collaborative scenario definition process and to identify requirements for a special-purpose GSS scenario tool.

The next section provides additional background and motivation for this research. Section 3 summarizes the research methodology used and defines measures for scenario quality and user productivity. Experimental results are presented in section 4 with their implications discussed in section 5. Research contributions and future research plans are summarized in the final section.

2. Background

2.1. Scenarios

Scenarios are narrative descriptions of the sequence of activities that a user engages in when performing a specific task. They are being increasingly used in both the human-computer interaction and software engineering communities throughout all phases of development [7] including requirements elicitation [8-12]. Because of the wide variety of use and context, there are several different types and formats for scenarios commonly used [13].

The software engineering community is probably most familiar with the interaction scenario, more often referred to as a "use case" [14]. This type of scenario focuses on the interaction of a user and the system necessary to accomplish a specific system-supported transaction [15].

Alternatively, "rich" [15] or "contextual" [16] scenarios are designed to capture a larger organizational context by describing all activities the user engages in to accomplish a business task (not just interactions with the system), as well as information about the user's business goals, resources required, and the social setting [15]. Contextual scenarios are especially useful during requirements elicitation when user groups seek to develop common definitions of business processes, assess opportunities for process improvement, and evaluate the business impacts of alternative system solutions. Pohl and Haumer [16] surveyed contextual scenarios modeling techniques to develop data models defining the information required at the scenario and interaction levels.

Formats of scenarios also vary widely. Textual formats are the most common, ranging from unstructured natural language [e.g., 7] to structured natural language such as tabular scripts [11] to formal grammars [10]. Various modeling techniques such as scenario trees [9], other graphical representations [e.g., 8], or prototypes may also be used instead of or in addition to textual descriptions.

A survey of current scenario practices in industry [12] reports that both contextual and interaction scenarios are being used and are primarily documented in a textual format using a word processor. A description of CSEM scenario usage is provided next.

2.2. Scenarios in CSEM

The primary purpose of scenario usage in CSEM is to support collaborative requirements elicitation. The goal is to use scenarios to support an evolutionary process that starts with rich descriptions of business processes, and then evolves those contextual scenarios into specific system use cases, prototypes, and finally into behavioral requirements specifications. This is accomplished through an iterative process of individual user scenario definition, group refinement of scenarios, development team analysis and implementation, and group review and feedback until behavioral requirements are agreed upon. Both the type and format of scenarios evolve as shown in the following overview of the CSEM requirements elicitation process.

1. Initially, users work in parallel to develop rich, contextual business scenarios to describe current work processes for each of the business activities with automation potential identified during high-level business activity modeling. CSEM contextual business scenarios are documented using textual descriptions, but no specific format has yet been defined. (Note: The current experiment only addresses this step. Future research will address the other steps.)
2. Then, the group is led to converge on a common definition of business processes through iterative refinement of the contextual scenario descriptions. Group members also brainstorm on how work changes or information systems might improve the business process and refine the business scenarios accordingly. Using scenarios to achieve group consensus, as is done in this step, is one of the most common uses of scenarios during requirements elicitation [12].
3. Scenarios are also used in conjunction with data modeling to assess how recommended data integration changes the business scenarios [17]. If the impacts are acceptable, the business scenarios are updated to reflect those changes. Otherwise, the user group recommends revisions to the integrated data model that are acceptable from a business processing perspective.
4. Next, the focus shifts from the business to the information system. Systems analysts, working with selected user representatives, use the business scenarios and data model to define alternative interaction scenarios, i.e., system use cases. Interaction scenarios are documented using a combination of text, graphical models, and screen mock-ups.
5. A prototype is developed based on the system use cases. The business scenarios are used to guide the prototype evaluation and provide a rich context for evaluating the impacts of alternative information systems solutions on the business. Group evaluation feedback is used to update the business scenarios, data models, use cases, and the prototype in an iterative development and evaluation process.

The scenario process is continuing to evolve based on user experiences like those described next.

2.3. Preliminary user experiences

Early CSEM scenario sessions with Department of Defense (DoD) user groups were extremely encouraging. A general-purpose GroupSystems tool, Group Outliner, was used to capture scenarios because it allowed structuring of the definition process through decomposition of business activities into scenarios and provided word processor-like capabilities for describing the scenarios. Facilitators observed that users found Group Outliner easy to use and quickly defined scenarios for a wide range of business activities based on general instructions about scenarios and their content. Because of initial variations in the type and level of detail of information included in scenario descriptions, more detailed instructions on data requirements were developed based on Pohl and Haumer's [16] contextual scenario data models. However, the format of the scenarios still varied greatly. For example, some scenarios were written as free-format textual descriptions, while others were clearly broken down into numbered steps. Users continued to ask for detailed format guidelines, but we were reluctant to provide them until we had a better understanding of the impacts of format on scenario quality and user definition productivity. This need led to the current research.

2.4 Current research

The current research seeks to respond to this need by focusing on the initial individual user definition of contextual scenarios using a textual format. The first step in the research was to develop a greater understanding of user-defined scenarios, i.e., *What is the quality of user-defined contextual scenarios?*

The next step in the research focused on comparing three of the most commonly used textual scenario formats. The scenario literature summarized previously showed the wide variety of formats used to describe scenarios. The three most common textual scenario descriptions were: (1) unstructured free-format narrative text, (2) separately numbered steps with free-format descriptions, and (3) more-structured tables with separate columns for each scenario data item and rows for each scenario action. However, little information is provided on why these formats were chosen or on the impacts of these formats. Therefore, the second research question is: *How will the format of scenario descriptions affect scenario quality and productivity of user definition of those scenarios?*

3. Research methodology

3.1. Experimental design

These research questions were explored using a laboratory experiment. During the experiment, a convenience sample of undergraduate MIS students used the same GroupSystems tool (Group Outliner) used in the DoD sessions to define scenarios for their university's current course registration system using one of the three common scenario textual formats.

Each experiment lasted approximately two hours and included a short Group Outliner training session, a practice exercise, definition of the course registration scenarios, and completion of a post-session questionnaire. Subjects spent 45 minutes defining up to four course registration scenarios. They were told to concentrate on developing the best possible description for the first scenario and that they did not need to complete all scenarios. If they finished the first scenario, they worked on the other scenarios. All subjects, regardless of treatment, where provided with an initial outline listing scenario categories (e.g., UA Course Registration Scenarios) and specific scenarios (e.g., Student Registers for Spring Semester Classes). The treatments varied based on what format subjects used to input the descriptions for those scenarios. All subjects were given a handout which provided (1) a common definition of what information should be included in a scenario description (Figure 1), (2) instructions on how to input the scenario description for their specific treatment format, and (3) a sample scenario using their treatment format.

To develop a scenario description, you must specifically identify each action that must be accomplished. ***For each action or step*** *in the scenario, you must identify* ***who performs that step*** *(e.g., a person or a system) and* ***what specific action is done****. You should also identify any* ***data/information that is needed*** *to complete that step, the* ***reason the step is done****, any* ***major error checking or exceptions*** *that may occur, as well as* ***alternative ways*** *of accomplishing that step.*

Figure 1 – Scenario experiment instructions

Subjects assigned to the first treatment entered their scenario descriptions as *unstructured* text into Group Outliner's comment window (Figure 2). Subjects assigned to treatment 2 listed the scenario actions as *numbered steps* by adding the actions as individual outline items in Group Outliner (Figure 3). Group Outliner automatically

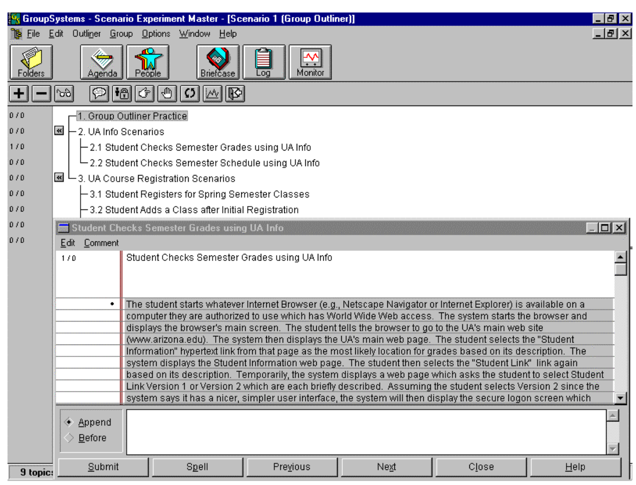

Figure 2 - Unstructured scenario format

generated step numbers. Finally, subjects assigned to treatment 3 defined *structured steps* by entering structured action names as individual outline items in Group Outliner and then adding detailed action information as comments (Figure 4). Group Outliner automatically generated step numbers.

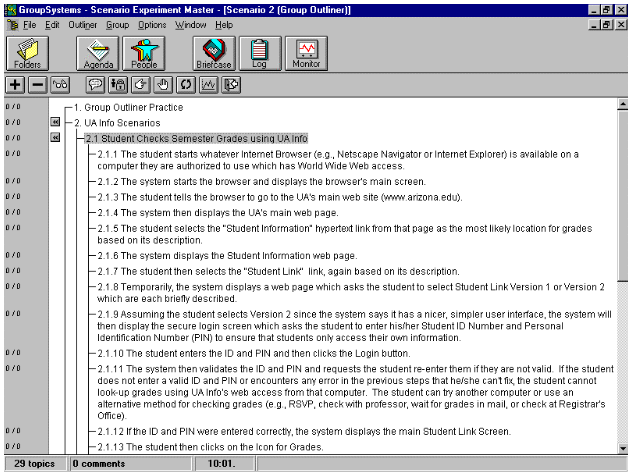

Figure 3 – Numbered step format

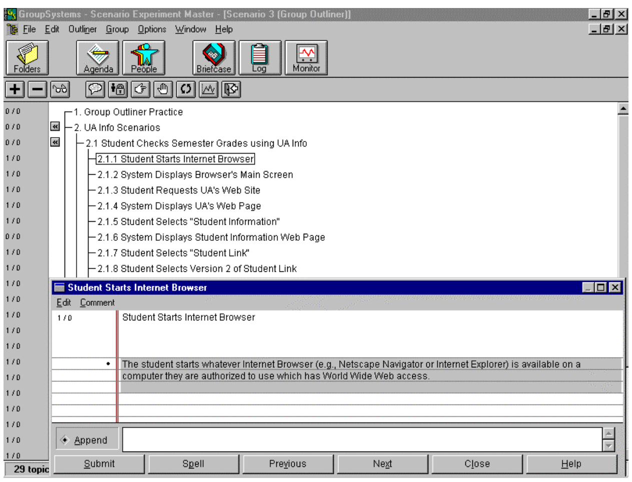

Figure 4 - Structured step format

The three formats served as the treatment levels in a single factor randomized complete block design. Experimental session was used as the blocking factor to control for (1) possible differences in experimental conditions between sessions and (2) the non-random assignment of subjects to sessions. Within each session, subjects were randomly assigned to treatment conditions. Seven experimental sessions with a total of 23 observations per treatment were conducted. The total sample size of 69 exceeded the 63 conservatively estimated as needed to achieve a power of .80 for α=.05.

3.2. Measuring scenario quality

Although the scenario literature identifies what information should be included in a scenario, it is essentially silent on how to measure scenario quality. The requirements literature identifies several measures for the quality of a software requirements specification (SRS) (see [18] for a summary). Four commonly mentioned are: completeness, correctness, understandability, and lack of ambiguity. Proposed scenario quality measures for these characteristics are discussed next.

Completeness. Davis et al. [19] state that an SRS is complete if "everything that the software is supposed to do is included in the SRS." Comparably, based on this experiment's definition of a scenario (Figure 1), a scenario is complete if every action that must be accomplished is fully described. Therefore, to judge scenario completeness, the total number of actors, actions, descriptions, data requirements, exceptions and alternatives were counted for each scenario and added together to calculate total scenario information.

Completeness can also be assessed at the individual action level, i.e., is all information about that action included in its definition. To judge action completeness, the percent of actions with actors, descriptions, data requirements, exceptions, and alternatives were calculated. The percentages were summed to get the average amount of scenario information provided per action to determine total action completeness.

Correctness. Davis et al. [19] define correctness at the SRS and individual requirement level with the latter measured as the percentage of individually correct requirements. In contrast, most processes developed to assess correctness (e.g., Fagan inspections) tally the number of defects or errors found during the inspection. Both approaches were explored in this research.

Understandability. Davis et al. [19] state that "an SRS is understandable if all classes of SRS readers can easily comprehend the meaning of all requirements with a minimum of explanation." However, directly measuring understandability is difficult and often very subjective. As an alternative, previous GSS experience has shown that group understanding of terms increases when definitions are provided. Therefore, for this experiment, the number and percent of action descriptions will be used as a surrogate objective measure of understandability.

Lack of Ambiguity. An SRS is said to be unambiguous "if and only if every requirement stated therein has only one possible interpretation" [18]. All scenario definitions were ambiguous because they were written in natural language with all its inherently ambiguities. Therefore, rather than quantifying this, specific ambiguous examples were highlighted in the scenarios to help identify the major sources of ambiguity so that future tools or techniques can seek to reduce them.

3.3. Measuring productivity

Productivity is generally measured as the quantity produced divided by the time or resources taken to produce that quantity. For scenarios, quantity has generally been measured as the total number of scenarios or the length of scenarios in lines or pages (e.g., see [12]). Since all subjects had the same time to define scenarios, the number of scenarios and scenario lengths will be used to measure of productivity. (Note: If subjects had worked different lengths of time, these totals would have been divided by subject's time.) Because of the variation in line and page sizes, length will be measured in words. It is not clear, however, whether scenario length is a valid measure of scenario quantity. For example, does a scenario description with 1000 words provide twice the information as one with 500 words or is it just more verbose? Also, does it take more words to express the same amount of scenario information in any of the formats? Therefore, before using word count to measure productivity, two preliminary analyses will be performed. First, the correlation between the length of scenario one and total scenario one information will be evaluated. Second, the ratio of word count to total scenario information will be analyzed to assess format conciseness. If correlation is high and no differences in conciseness are evident, then total scenario length for all scenarios will be used to measure overall productivity.

3.4. Questionnaire development

The post-session questionnaire was designed to collect:

1. Subject’s assessment of the quality of their scenarios and process productivity. The thirteen quality and six productivity questions directly paralleled the proposed quality and productivity metrics.
2. Ease of use of the tool used to define scenarios (GroupSystems Group Outliner) to ensure that it did not negatively impact scenario definition. These seven questions were directly taken from Fred Davis’ widely used perceived ease of use instrument [20].
3. Demographic data so that homogeneity of subjects across treatments could be assessed.
4. Self-reports on motivation to ensure that there were no significant differences between treatments.

3.5. Analytical techniques

Analysis of the experimental results was based on quantitative analysis of scenario quality and productivity measures and post-session questionnaires plus qualitative assessment of the experiments and scenario descriptions.

Analysis of scenario content focused on the first scenario since that was the only one defined by all subjects. To ensure accurate and consistent counts of the scenario content, a master spreadsheet was developed which consolidated all scenario information (e.g., actors, actions, descriptions, data requirements, exceptions and alternatives) provided by the subjects for scenario one. The spreadsheet was compared to the university’s course registration instructions to ensure that it accurately reflected the current process. Each scenario definition was then compared against the master spreadsheet to count the number of actions, actors, descriptions, data requirements, exceptions, and alternatives identified for that scenario. The other scenario quality and productivity measures were calculated from these counts. The total number of scenarios and words for each scenario were also counted.

Preliminary analysis focused on the demographic and motivation questionnaire results. A common factor analysis of the individual scenario quality, productivity, and ease of use questions was conducted to ensure the questions loaded on the appropriate factors. Factor scores were used to compare treatments on the resulting factors.

Quantitative analyses of the scenario quality and productivity measures began with evaluation of descriptive statistics followed by comparisons between treatments. Since experimental session was not significant as a blocking factor for any measure, detailed analyses were conducted using a one-way ANOVA for the treatment factor only.

4. Results

In general, all participants easily defined the main actions for each scenario. However, many did not provide all the requested information for each action (e.g., data requirements, exceptions, and alternatives). Problem areas and quality and productivity differences between the treatments are highlighted in the following sub-sections.

4.1. Participant demographics and motivation

Analysis of the demographic questions showed that no significant differences existed between treatment groups on these characteristics. Subjects were primarily senior MIS majors with multiple MIS courses, but only limited analytical expertise. On average, they used computers one or more times a day and had very good computer and word processing expertise with better than average typing skills. They also were somewhat familiar with GroupSystems. The other experimental check was for motivation and effort. On average, motivation and effort were very high (4.28 and 4.17 on a 5-point scale) with no significant differences between treatment levels.

4.2. Assessing scenario quality and productivity

The first research question was: *What is the quality of user-defined contextual scenarios?* To address this question, descriptive statistics for scenario quality were analyzed. All statistics showed a high degree of individual variability and identified important quality problems. For example, while the average number of actions per scenario definition was 18.62, the individual counts varied from a low of 6 to a high of 38. Most other measures showed similar variability. These results were consistent with observations during the experiments where some subjects seemed to be rapidly defining scenarios while others seemed to be struggling. Statistics on action completeness were also interesting and surfaced some potential quality problems. For example, while 95% of action specifications included the actor, less than 50% provided a description of the action with even fewer identifying data requirements (22%), exceptions (17%), or alternatives (4%). If these numbers are low for all treatments, then the scenario format, process, or tool must change to improve action completeness. Descriptive statistics for productivity measures showed similar high individual variability.

4.3. Comparing scenario quality and productivity

To respond to the second research question, *How will the format of scenario descriptions affect scenario quality and productivity of user definition of those scenarios*, this section compares the three treatment formats and the unstructured (treatment 1) and structured formats (treatments 2 and 3) on each of the scenario quality and productivity measures.

Completeness. The scenario completeness measures (Table 1) show slightly more actions and actors for treatment 1 than the other treatments, but these differences are not statistically significant. The higher numbers of descriptions, data requirements, and total scenario information for treatment 1 are weakly significant ($p < .10$) with the differences in numbers of alternatives significant at the .05 level. Comparing the unstructured and structured treatments increases the statistical significance of the differences to the .05 level for the numbers of descriptions, data requirements, alternatives, and total scenario information.

Fewer differences are significant when analyzing action completeness. As shown in Table 2, the difference in the percentage of actions with alternatives is weakly significant when comparing the unstructured and structured treatments. When viewing total action completeness as the average amount of information per action (maximum of 5), the differences between the unstructured and structured treatments is significant at the .05 level, with the differences between all treatments even more strongly significant ($p < .01$).

Correctness. As expected, there were no significant differences between treatments for any of the correctness measures. A qualitative assessment of the errors showed that many of the errors were caused by incorrect action sequence. However, these errors often identified viable alternative action sequences or highlighted areas where the action sequence was not very logical.

Understandability. The number and percent of action descriptions were used as surrogate measures for scenario understandability. As shown in Table 1, there is weak statistical evidence that treatment 1 scenario definitions are more understandable when comparing all treatments and slightly stronger evidence when comparing the unstructured to the structured treatments, based on the higher number of descriptions provided for treatment 1. Although there are also differences between the percentage of actions with descriptions between treatments (Table 2), these differences are not significant because of the high degree of individual variability.

Lack of Ambiguity. As planned, assessment of scenario ambiguity was qualitative. The most common problem area was vague specification of data requirements. For example, many subjects used course number and course call number interchangeably in their scenario definitions when in fact they are distinctly different data items with totally different formats. Another common problem area was ambiguous specification of actors as he/she/it. For example, sometimes it was not clear whether the actor was the student, instructor, or registrar personnel.

Productivity. The final measures related to subject user productivity when developing scenarios. As shown in Table 3, there were no significant differences in the total number of scenarios defined in the allotted time frame. However, subjects using the unstructured format clearly used more words to define scenario 1 and more total words for all scenarios defined.

Since the correlation between the number of words and the total information in scenario 1 was .80 and there were no statistically significant differences in conciseness between treatments (see Table 3), the total word count can be used as a valid estimate of productivity. As Table 3 shows, the total word count was significantly higher for the unstructured format when comparing the three treatments and the unstructured and structured treatments.

4.4. Analysis of post-session questionnaire results

To further explore the second research question, subject's perceptions of scenario quality, productivity, and ease of use, collected as part of the post-session questionnaire, were analyzed. The analysis showed that subjects generally rated scenario quality, productivity, and GroupSystems ease of use very high. There were no significant differences between treatments for scenario quality or GroupSystems ease of use. In contrast, subjects rated the structured treatments significantly higher on (1) the ease of the scenario definition method, and (2) whether the method allowed them to do what they needed.

Although the questionnaire results showed few treatment differences, the results did seem to indicate consistency between related questions. Results of the common factor analysis performed to explore this commonality are summarized in Table 4. (Detailed factor scores are available on request, but were not included in because of space constraints.) Significance tests reported using the maximum likelihood method of factor analysis with varimax rotation indicated that three factors were sufficient. These factors directly map to the quality, ease of use, and productivity concepts. In addition, the Cronbach's alpha for all factors met or exceeded the recommended .80 standard for business research.

Means of standardized factor scores for each treatment and the unstructured versus structured treatments are summarized in Table 5. As expected from the individual question analysis, the only statistically significant difference is between the unstructured and structured treatments with the structured groups rating productivity higher than the free-format group.

5. Discussion

The experimental results can be summarized into three main findings: (1) scenario completeness was low for all formats, (2) completeness was lower for structured formats than the unstructured format, and (3) results of higher productivity measures for the unstructured format contradicted subjects' perceptions of higher productivity for structured formats. Possible reasons for each of these findings is discussed in the next section. Limitations of the study and their potential impact are summarized in section 5.2. Finally, implications of the findings and limitation on the scenario definition process and GSS tool are discussed in sections 5.3 and 5.4.

5.1. Interpretation of results

The results clearly showed completeness problems for all formats at both the action and scenario level. At the action level, only half of actions included descriptions, with less than a quarter including data requirements, exceptions, and alternatives. One possible explanation is that, although the instructions requested this information, none of the formats specifically prompted for it, so subjects had no on-screen reminders as they developed the scenario descriptions. In addition, subjects were asked to provide all the information at once, which may have caused an information overload problem. Most subjects did well identifying the main actions, but may have done better with total action completeness if the definition process had been split into at least two steps: (1) identifying the actions and then (2) adding the remaining detailed action information. At the scenario level, completeness was also low with only an average of 18.6 actions included per scenario out of a total 65 unique actions identified by one or more subjects. One possible cause for this problem was the fact that subjects developed scenario definitions individually and did not collaborate to capture actions identified by others.

In addition to the common completeness problems, completeness was significantly lower for the structured formats when compared to the unstructured format at both the scenario (see Table 1) and action (see Table 2) level. This may have been caused by the structured formats' action numbering which increased focus on the action sequence at the cost of lost attention to the other action information. Format 3's named and numbered steps magnified this problem by totally separating the action from the other information. Both structured treatments' focus on sequential actions also made it more difficult to represent non-sequential information such as exceptions and alternatives. Most importantly, the structured treatments provided no specific structural support for increasing completeness, e.g., with prompts or templates.

Productivity results were somewhat contradictory, with the productivity metrics clearly indicating that the unstructured format was more productive (see Table 3) while subjects felt more productive using the structured formats (see Table 5). One possible explanation for this contradiction is that while the unstructured format didn't hinder productivity, it also didn't provide any specific support for scenario definition. So when subjects were asked if the method they used supported scenario definition, they preferred the structured formats which used more of Group Outliner's capabilities. In addition, subjects may have focused on the higher word count of the unstructured format without recognizing the associated increased content, thereby feeling less productive because more words were required for their scenario definitions.

Finally, in interpreting the results it should be recognized that although the results showed many differences between treatments, the majority of these differences were not statistically significant or were only weakly significant. There were two primary causes for this lack of statistical significance. First, the high individual variability outweighed differences between treatments, making the probability of detecting true differences extremely low. This variability was accounted for in the statistical calculations, but in the future it may be appropriate to increase sample size or seek methods to reduce the individual variability to increase the power of the statistical tests. Second, the individual treatments themselves may not have been distinct enough to cause significant differences. This problem was partially addressed in this paper by comparing the unstructured treatment to both structured treatments (2 and 3). Future research should increase the distinction between treatments.

5.2. Limitations of the study

Some of the variability in the scenario descriptions may have been caused by vague definition of the scope of the experiment's scenarios. For example, while some subjects felt registering for classes started with calling the university's course registration system, others began with planning their schedules. Still others began with determining the courses they needed to take based on their major. While this may be useful when trying to develop an initial understanding of the complexities of a scenario, in general it will make reaching consensus much more difficult. In the future, the scope of scenarios should be more clearly defined before beginning definition of the actions necessary to accomplish that scenario.

A second limitation of this study was that, although subjects were told what to information to include in a scenario definition and how to document that information using Group Outliner, they were not told how to actually develop the scenario definition. This lack of process guidance may have been a major contributor to the action completeness problems observed. For example, the action completeness results (Table 2) seem to indicate that subjects focused on identifying the main actions only. If they had been told to identify the main actions first, and then go back to add the other action information, action completeness may have been much higher. Future research should evaluate how an improved process can increase scenario quality and productivity.

A final limitation of the study was that the combination of the selected scenario formats and the Group Outliner tool may have significantly limited subjects' ability to quickly define quality scenarios. As a minimum, neither seemed to provide any support for any of the completeness problem areas. Future research should investigate alternative scenario format and tool combinations to improve quality and productivity.

5.3. Scenario definition process implications

The most important implication of the results and the first two limitations is that an iterative scenario definition process that incorporates collaborative review between process steps is needed. An iterative process should increase clarity of scenario scope, reduce the information overload problem and may improve scenario completeness as each step focuses on a specific sub-set of the required scenario information. A collaborative process enables information sharing which may also increase completeness. Experimental observations and the ranking of action completeness results (Table 2) implies the need for a collaborative scenario definition process such as:

1. Users should first define and agree upon scenario goals and preconditions to ensure a common understanding of the scope of each scenario.
2. Next, users should work in parallel to identify the normal action sequence for all scenarios, followed by group review and refinement.
3. Once the actions are agreed upon, users can add additional action details such as more complete action descriptions and data requirements.
4. Finally, as required, users can add critical exceptions or error checking and identify alternative ways of accomplishing actions to complete the scenario definition.

5.4. GSS scenario tool implications

The results and the final limitation also highlighted weaknesses in using Group Outliner for scenario definition. Although Group Outliner supported structuring scenarios into named and numbered actions, this structure did not provide sufficient support for developing complete scenario definitions. Providing specific prompts or a template that could be completed in an iterative manner should improve scenario completeness, but its impact on productivity must be evaluated. A template could be used with Group Outliner, but it would be somewhat awkward and inflexible, increasing the possibility of negative productivity impacts. Alternatively, a GSS scenario tool could be specifically designed to provide the necessary information prompts with added quality and productivity aids such as pull-down lists to reduce ambiguity (e.g., for actors and data), and targeted support for iterative scenario definition.

6. Conclusion

Scenarios have great potential for helping to solve many of the requirements problems that have been plaguing the software industry since its inception. Their primary value centers on their simplicity – users can easily describe concrete examples of what and how they do their jobs. These examples provide a rich source of information for aiding the discovery of requirements and for evaluating alternative ways for meeting those requirements. However, when used in an undisciplined manner, scenarios can also be incomplete, unfocused, fail to provide critical information, and extremely inefficient to define and analyze. The purpose of this research was to take a first step and set a foundation for a disciplined collaborative scenario elicitation process that can accrue the benefits of scenarios while avoiding their pitfalls. This research's primary contributions is in its identification of the quality weaknesses of currently used scenario formats, processes, and tools and its definition of new format, process, and tool concepts that should be evaluated in the next step in research to create a disciplined collaborative scenario elicitation process.

Practitioners planning to incorporate scenarios into their requirements elicitation process now should consider implementation of an iterative, collaborative scenario elicitation process using a standardized scenario template. Based on the findings from these experiments, we have begun to use this approach with DoD user groups and have seen some highly encouraging results. Using scenario templates with GroupSystems Group Outliner has noticeably increased the quality and consistency of scenario definitions. However, Group Outliner's limited support for templates has led to development of a GSS scenario prototype which can fully take advantage of the lessons learned during this experiment.

Future research will focus on the formal evaluation and enhancement of the proposed collaborative scenario elicitation process and prototype. Plans are to refine the individual scenario definition format, process, and tool and then move on to collaborative scenario definition, refinement, and evaluation research that more fully supports all aspects of scenario use in the Collaborative Software Engineering Methodology.

References

[1] Standish, "Chaos," Standish Group International, Inc., Dennis, MA 1995.

[2] B.W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[3] J.F. Nunamaker, Jr., R.O. Briggs, D.D. Mittleman, D.R. Vogel, and P.A. Balthazard, "Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings," *Journal of Management Information Systems*, Vol. 13, No. 3, 1996-97, pp. 163-207.

[4] D.L. Dean, J.D. Lee, M.O. Pendergast, A.M. Hickey, and J.F. Nunamaker, Jr., "Enabling the Effective Involvement of Multiple Users: Methods and Tools for Collaborative Software Engineering," *Journal of Management Information Systems*, Vol. 14, No. 3, 1997-98, pp. 179-222.

[5] D.L. Dean, J.D. Lee, R.E. Orwig, and D.R. Vogel, "Technological Support for Group Process Modeling," *Journal of Management Information Systems*, Vol. 11, No. 3, 1995, pp. 43-64.

[6] J.D. Lee and D.L. Dean, "Tools and Methods for Group Data Modeling: A Key Enabler of Enterprise Modeling," *ACM SIGOIS Bulletin*, No. August, Special Issue on Enterprise Modeling, 1997

[7] J.M. Carroll, *Scenario-Based Design: Envisioning Work and Technology in System Development*, John Wiley & Sons, Inc., New York, 1995.

[8] K.M. Benner, M.S. Feather, W.L. Johnson, and L.A. Zorman, "Utilizing Scenarios in the Software Development Process," *Proceedings of the IFIP WG 8.1 Conference on Information System Develop Process*, Elsevier Science Publishers B.V., North-Holland, 1993, pp. 117-134.

[9] P. Hsia, J. Samuel, J. Gao, D. Kung, Y. Toyoshima, and C. Chen, "Formal Approach to Scenario Analysis," *IEEE Software*, Vol. 11, No. 2, 1994, pp. 33-41.

[10] J.C.S.d.P. Leite, G. Rossi, F. Balaguer, V. Maiorana, G. Kaplan, G. Hadad, and A. Oliveros, "Enhancing a Requirements Baseline with Scenarios," *Proceedings of the Third International Symposium on Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 44-53.

[11] C. Potts, K. Takahashi, and A.I. Anton, "Inquiry-Based Requirements Analysis," *IEEE Software*, Vol. 11, No. 2, 1994, pp. 21-32.

[12] K. Weidenhaupt, K. Pohl, M. Jarke, and P. Haumer, "Scenarios in System Development: Current Practice," *IEEE Software*, Vol. 15, No. 2, 1998, pp. 34-45.

[13] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyte, A. Sutcliffe, N.A.M. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, and P. Heymans, "A Proposal for a Scenario Classification Framework," *Requirements Engineering Journal*, Vol. 3, No. 1, 1998, pp. 23-47.

[14] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, Reading, MA, 1992.

[15] K. Kuutti, "Work Processes: Scenarios as a Preliminary Vocabulary," *Scenario-based design: Envisioning work and technology in system development*, J. M. Carroll, ed., John Wiley & Sons, Inc., New York, 1995.

[16] K. Pohl and P. Haumer, "Modeling Contextual Information About Scenarios," presented at Third International Workshop on Requirements Engineering: Foundation for Software Quality, Barcelona, Spain, 1997.

[17] A.M. Hickey, D.L. Dean, and D.R. Vogel, "Participative Analysis of Systems Integration Opportunities," *Proceedings of the Americas Conference on Information Systems*, Association for Information Systems, Indianapolis, IN, 1997.

[18] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and Measuring Quality in a Software Requirements Specification," *Software Requirements Engineering*, R. H. Thayer and M. Dorfman, eds., 2nd Ed, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 164-175.

[19] A.M. Davis, *Software Requirements: Objects, Functions, and States*, Revised Ed, Prentice Hall PTR, Upper Saddle River, NJ, 1993.

[20] F. Davis, "Perceived Usefulness, Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly*, Vol. 13, No. 4, 1989, pp. 319-340.

Table 1 – Mean scenario completeness measures

Scenario Completeness	Treat 1	Treat 2	Treat 3	Unstructured (Treat 1)	Structured (Treat 2&3)
# Actions	19.91	17.78	18.17	19.91	17.98
# Actors	19.00	17.00	17.30	19.00	17.15
# Descriptions	**10.78***	**8.78***	**7.96***	**10.78****	**8.37****
# Data Requirements	**5.04***	**4.09***	**3.26***	**5.04****	**3.67****
# Exceptions	3.22	2.65	2.04	3.22	2.35
# Alternatives	**1.13****	**0.65****	**0.43****	**1.13****	**0.54****
Total Scenario Information	**59.09***	**50.96***	**49.17***	**59.09****	**50.07****

* Weakly Significant (p <.10), ** Significant (p<.05)

Table 2 – Mean action completeness measures

Action Completeness	Treat 1	Treat 2	Treat 3	Unstructured (Treat 1)	Structured (Treat 2&3)
% Actions with Actors	0.95	0.95	0.95	0.95	0.95
% Actions with Descriptions	0.53	0.50	0.43	0.53	0.46
% Actions with Data Rqmts	0.25	0.23	0.18	0.25	0.21
% Actions with Exceptions	0.16	0.14	0.11	0.16	0.12
% Actions with Alternatives	0.06	0.04	0.03	**0.06***	**0.03***
Average Info per Action	**1.96*****	**1.86*****	**1.70*****	**1.96****	**1.78****

* Weakly Significant (p <.10), ** Significant (p<.05), ***Strongly Significant (p<.01)

Table 3 – Mean productivity measures

Productivity	Treat 1	Treat 2	Treat 3	Unstructured (Treat 1)	Structured (Treat 2&3)
Total # Scenarios	2.61	2.61	2.43	2.61	2.52
# Words for Scenario 1	**501.35****	**421.00****	**358.83****	**501.35****	**389.91****
Conciseness for Scenario 1	12.28	12.38	11.35	12.28	11.87
Total # Words, All Scenarios	**800.30****	**632.48****	**567.30****	**800.30*****	**599.89*****

** Significant (p<.05), ***Strongly Significant (p<.01)

Table 4 - Questionnaire factor analysis summary

#	Factor Description	Proposed Items	Actual Items	Cronbach's Alpha
1	Quality	13	11*	.89
2	Ease of Use	7	8**	.90
3	Productivity	6	5**	.80

* Questions on alternatives and exceptions dropped due to low commonality with other variables.

** Because of vague question wording, one productivity question loaded slightly higher on ease of use.

Table 5 - Comparison of questionnaire factor scores

#	Factor Description	Treat 1	Treat 2	Treat 3	Unstructured (Treat 1)	Structured (Treat 2&3)
1	Quality	0.10	0.14	-0.25	0.10	-0.05
2	Ease of Use	-0.17	0.16	0.00	-0.17	0.08
3	Productivity	-0.28	0.08	0.20	**-0.28****	**0.14****

** Significant (p<.05)