Engagement vs. Reflection: A Study of User Interfaces for an Intelligent Story Authoring Assistant

Brian O’Neill
School of Interactive Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
boneill@cc.gatech.edu

Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
riedl@cc.gatech.edu

ABSTRACT
As a desire to create media for consumption has increased, tools have been developed to aid in the creative process for several forms of media. However, creating content that will be considered valuable by the community is challenging, and there remains a need for tools that support novices. We approach this problem with ReQUEST, an intelligent story authoring assistant that acts as a surrogate audience, providing feedback as the user writes. In this paper, we describe a study comparing two interfaces, differing in the visual representation of the story and the presentation of feedback to the user. Users of one interface appeared to enter into a reflection phase of the creation process more often. However, this interface may have encouraged too much reflection, possibly leading users to revise their stories despite believing that it would not improve their creation.

Author Keywords
Intelligent Authoring Support, Story Authoring, User Interfaces, Mixed-Initiative Systems

ACM Classification Keywords

INTRODUCTION
A growing trend in the world is the desire to create media, rather than simply consume. Today, tools exist for creating nearly every media imaginable. However, the mere ability to create does not guarantee that a particular community will value the new work, or respect its quality. Because of this gap between simply creating and creating with value, tools are needed to support non-expert authors in their efforts. Tools that support and scaffold the creation of content would be more valuable than a tool that simply manipulates data.

Story telling is an integral aspect of many forms of media, from literature to movies, and their non-expert equivalents such as fan-fiction and machinima. Consequently, in this paper we address support for story authoring. In particular, we focus on the plot of the story, for the time-being, leaving the user to address other elements of storytelling, such as character dialogue, for later in the process. This approach is influenced by McKee’s [7] description of the screenwriting process, which begins with the plot in prose, before adding elements inherent to screenwriting, such as camera angle, character placement, and dialogue.

Further, this approach leads us to our long-term goal of a mixed-initiative system [3], capable of providing feedback and solving problems in conjunction with the user on a number of activities related to content creation. Artificial intelligence support of creative content authoring differs from mixed-initiative support of problem solving in subtle ways. Lubart [6] enumerates four ways in which computer interfaces can support creativity:

- **Computer as nanny:** The computer provides organizational and classification services and performs routine operations on behalf of the user.
- **Computer as pen-pal:** The computer facilitates brainstorming with functionality that captures and transmits to collaborators the user’s thoughts.
- **Computer as coach:** The computer is knowledgeable about the process and can offer suggestions and stimulate creativity.
- **Computer as colleague:** The computer forms half of a human-computer team by contributing to the solution.

The computer as coach metaphor is used extensively in intelligent tutoring systems [16]. However, it is not our intention to teach a user how to author plots. A system-as-coach acts to facilitate improved task performance by a user, but does not attempt to solve or contribute to the problem on which the user is working. In contrast, the computer as colleague metaphor introduces automation into the creation process. Typical mixed-initiative support systems implement an expert system that is capable of solving some, or all, of the problem the human user is working on. The user and the system take turns, filling in details of the solution.
Our approach to computational support for creativity is Re-QUEST; an intelligent story authoring support system. Re-QUEST does not act as an expert on story authoring, as a typical mixed-initiative support system might, but rather as a surrogate audience. We see this computer-as-audience role falling between Lubert’s coach and colleague positions. Re-QUEST takes elements of the computer-as-coach perspective, assisting novices in the writing process. This assistance comes not from suggesting or providing plot elements, but rather from acting as a reader, giving feedback to the author by asking questions about the story-so-far. Acting as a model of the reader also places Re-QUEST partially in the computer-as-colleague approach. While it does not directly contribute to the solution, Re-QUEST is still providing the author with valuable feedback that is more than a suggestion. This computer-as-audience approach fits with existing creative design models, such as Sharples’ [15] model of engagement and reflection, by providing intermittent opportunities for an author to create and receive feedback.

How an intelligent agent interacts with the human user can be just as important as how it functions. For example, many mixed-initiative systems model interaction as dialogue [3]. In this paper we investigate possible user interfaces for story authoring with Re-QUEST as a back-end. Each has implications for the visible representation of the story, how system feedback is presented to the user, and ease of integrating artificial intelligence into the user interface.

This paper begins with a description of Re-QUEST, including an overview of the QUEST model and Re-QUEST algorithm. This is followed by a description of our two interfaces and the user interface study we conducted. The next sections contain results and discussion of the study, specifically pertaining to participant actions in creating events and answering questions, including quotes gathered in our interviews with subjects. Future work and related work follow, with a focus on past narrative authoring and narrative understanding systems.

**REQUEST**

Re-QUEST is an intelligent story authoring assistant that aids novice writers by modeling a story as it is authored and providing feedback about that story. Re-QUEST serves as a model of a reader, keeping a graphical data structure of the story-so-far as it would be interpreted by a reader. The model of the story is based on that used by the QUEST question-answering technique. Re-QUEST uses the QUEST model of question-answering to identify gaps in reader understanding and to ask questions to fill those gaps. As the author creates the story, the Re-QUEST algorithm attempts to resolve unclear aspects of the story. Re-QUEST does this in much the same way as a human reader, querying its memory of story events for possible explanations. The algorithm itself does not understand what it reads, but tries to make sense of the story based on causal and character-goal structural relationships. Because the story is incomplete at most times during authoring, the Re-QUEST reader model may be incapable of making these connections, and therefore poses questions to the user in order to draw out these connections.

The questions also serve to help the author flesh out details of their story that may have been glossed over. Thus Re-QUEST serves as a sounding board—an opportunity for an author to see, anticipate, and resolve questions that a future human reader might ask.

The remainder of this section describes QUEST and the fundamental elements of its model, followed by the rules of the Re-QUEST algorithm, and how it uses questions posed to the writer.

**QUEST Model**

QUEST [4] is a psychological model of question answering that simulates the question-answering performance of humans when responding to open-class questions about narrative content. In particular, QUEST models the answers to several classes of questions typical of human readers, including causality, consequence, and why questions, among others. QUEST represents stories as directed graphs, with nodes representing plot elements and character goals, while arcs represent relationships between elements.

The QUEST model contains three types of nodes and four types of arcs that apply to the Re-QUEST algorithm. The three node types that relate to Re-QUEST are Events, States, and Goals:

- **Event nodes** represent the actions that take place in the story. The contents of Event nodes come directly from what the user writes.
- **State nodes** describe the status of the world in the story. Like Events, these nodes are formed from text input by the user.
- **Goal nodes** are created for each event intentionally performed by the characters in the story. They represent the goals, of varying importance, of the characters in the story.

The prominent QUEST arc types used by Re-QUEST include Consequence, Reason, Initiates, and Outcome arcs. All arcs in the QUEST model are directed.

- **Consequence (C) arcs** denote causality or enablement between states or events. The source event causes the target event, or enables its occurrence.
- **Reason (R) arcs** connect goals, and suggest that one goal could be considered a sub-goal of another. A chain of goals linked by Reason arcs suggest a plan being carried out by one or more characters. When combined with a sequence of events connected by Consequence arcs, such a chain is referred to as a goal hierarchy.
- **Initiates (I) arcs** connect an event to a goal. Such an arc indicates that the event led a character to adopt the goal. When an event initiates a goal that is part of a goal hierarchy, the Re-QUEST algorithm assumes that the initiated goal is the overarching goal of the hierarchy.
- **Outcome (O) arcs** connect goals to events. These arcs imply that the action taken in the event was the outcome
of having a particular goal, and thus the achievement of that goal.

To predict human question answering behavior, QUEST traces the graph using traversal routines for each type of question. While these routines inform ReQUEST processing, a detailed discussion is outside the scope of this paper.

Figure 1 shows a sample QUEST data structure. The diagram depicts a short portion of a story called “The Czar’s Daughters” [4]. In the story, the czar’s three daughters are kidnapped by a dragon and eventually rescued by heroes. In the portion depicted, the heroes hear the daughters cries, initiating the goal of rescuing the girls. In order to rescue the daughters, the heroes have to go to the dragon and fight it. As a consequence of fighting the dragon, the daughters are rescued. The sequence of events and goals depicted is an example of a goal hierarchy. The full extent of the QUEST model is beyond the scope of this paper. A more complete explanation the QUEST model can be found in [4].

ReQUEST Algorithm

The ReQUEST algorithm keeps a QUEST model of the story as written so far by the author. When the model is unable to resolve gaps between elements of the data structure, questions are presented to the user. There are two types of questions in the ReQUEST algorithm:

- “Why” Questions seek the cause of their associated event or state. Once answered, events or states may be linked to each other by Consequence arcs and goals by Reason arcs. Why questions may also be associated with events that have no goals. In this instance, the associated event typically is linked to a goal hierarchy by an Initiates arc.

- “Consequence” Questions seek the resulting action from some event or state. Once answered, states and events are linked to each other by Consequence arcs. Consequence questions typically take the form “What was the consequence of [some event]?”

The algorithm poses a question when an event is created and it is not (1) part of a goal hierarchy, or (2) the terminus of a Consequence arc. Questions are also created for state events that are not the consequence of some other state or event. This may prevent some nonsensical answers. Questions are answered when (1) the associated event becomes part of a goal hierarchy, or (2) the associated goal is connected to an event by an Initiates arc, or (3) the event is the consequence of some other event or state. Generally, answering a question results in at least one of these three possibilities, however a nonsensical or self-referential answer may not.

The algorithm itself does not provide any rules for the wording of questions. This task is left to the user interface. Further, ReQUEST does not attempt to perform any natural language understanding, relying completely on the relationships between events and goals that can be established automatically or through question-answering. In the ReQUEST algorithm, all outstanding Why questions must be resolved before any Consequence questions are posed to the user. While ReQUEST, as an algorithm, is capable of operating behind the scenes, the human author requires a user interface that facilitates manual authoring of plot events and also provides a modality for the user and the ReQUEST agent to exchange information.

USER INTERFACES

This paper presents the results of a study comparing two potential user interfaces for ReQUEST. Although several possible interfaces were considered, we chose the two described below in order to focus on differences between particular elements of the UI. Specifically, what arrangement of the questions would be least distracting? Could authors understand and write within a simplified data structure? Would being given a particular axis of time assist or constrain the user?

Whiteboard Interface

The first interface is designed to reproduce a naturalistic style of story plotting, mimicking an index-card structure popular with writers, especially screenwriters. For the purposes of this study, this interface (Figure 2) was run on a whiteboard with Post-It notes. In this interface, we chose to distance questions from plot content material, placing them on a second board next to the main whiteboard. Questions were answered by indicating that a certain question was being addressed by a particular event on the board. When questions were answered, they were marked as such on the second whiteboard. Additionally, the whiteboard provided an im-
Subjects were asked to tell a story about a graduate student who experiences some sort of betrayal. They were instructed that in telling the story, they could only provide the events of the story, as well as provide state information. Subjects were also informed that as they added events, the Wizard would present questions pertaining to their events. Participants were told that they could choose to ignore questions for as long as they liked, including opting to never answer a question. They were then told that there were only two ways to answer questions: (1) Answering with an existing event, or (2) Creating a new event and answering with that one. Participants in the whiteboard case were asked to read new events aloud and to verbally indicate when they were answering a question and how they were doing so. Throughout the process, the Wizard kept a ReQUEST data structure on paper, which was used to model the story and determine questions, acting in place of an artificial intelligence running the ReQUEST algorithm. A timed log of actions taken by users were kept in both cases, on paper for the whiteboard case, and through logging actions in the computer case. Photographs or screenshots were taken of each participant’s final story. Following the completion of the participant’s story, a short interview covered opinions of the interface and the algorithm, past writing experience, and particular actions taken by the user during the writing process.

In the computer condition, users were given a short opportunity to practice using the interface before beginning to tell their stories. This practice provided participants with the opportunity to attempt each of the tasks they would have to complete, specifically creating new events, linking them with an edge, and seeing what a question would look like. The Wizard, sitting at a laptop across from the user, used a special Wizard interface that allowed him to add questions to the participant’s screen.

Interview

Once subjects had finished their stories, they were asked to continue their participation with an interview. Each interview began with a short set of questions meant to gauge the participant’s experience with writing and telling stories. Afterward, subjects were asked what they thought of the tool they had just used, what they had found difficult, and what might have made the writing process easier. Subjects were also asked to describe what they thought about how they added events and how they decided when to answer questions. After answering general questions about the interface, participants were asked questions about their particular actions during their storytelling process, such as answering a question with multiple events or seemingly placing events outside the timeline.

RESULTS

Our analysis compared both subjects’ use of the system as well as aspects of the stories they created. In general, data was compared on a between-group basis where each group interacted with a different interface. Stories were compared based on number of events and the form of the data structure, while users were compared on the frequency of answering questions and specific actions in dealing with questions. Stories were also read by an independent group of judges.
to determine which interface produced better stories. Due to the small sample size, none of our data was found to be statistically significant. However, we believe we have identified interesting trends in the data that suggest implications for future work on intelligent creativity support systems.

**Answering Questions**

One intriguing aspect of the results was how each group handled the questions posed by the ReQUEST algorithm. Subjects in both conditions were given questions on approximately the same percentage of events created (78.3% in Whiteboard; 76.3% in Computer). However, the two groups differed in the rate at which they answered questions, and how they chose to answer questions. Subjects in the whiteboard interface condition on average answered 58.9% of the questions posed, while those using the computer interface answered 72.9% of questions. Further, we tracked whether subjects answered questions with events that already existed or by creating new events. When answering questions, subjects in the whiteboard interface condition answered with new events 55.8% of the time, while subjects in the computer interface condition created new events to answer questions 76.9% of the time. (See Table 1)

**"Backchaining" Within Stories**

Related to the users’ process of answering questions was their tendency to create a new event which, within the context of the story, occurred temporally before a pre-existing event. While only 20.8% of events in the whiteboard interface condition were created under such conditions, 35.1% of events in the computer interface condition were authored to precede a pre-existing event in the story’s timeline. In both cases, most events added in this manner were created to answer a question. These events were used to answer questions 55.0% of the time in stories from the whiteboard interface, while they were used 67.9% of the time in stories created in the computer interface.

We define a backchain to be a sequence of two or more connected events in the data structure that were created in reverse temporal order and causally connected to a pre-existing event that occurs within the story after the events in the backchain. We require two events, because a single event that meets these requirements could simply be a minor modification or correction caused by answering a question. Furthermore, we define a tangent to be a backchain that does not connect to the rest of the story through an event other than the pre-existing event linked to by the backchain. Typically, tangents and backchains were goal hierarchies; however, in some instances, they could be initiated by a single event or state. Figure 4(a) shows a backchain in a QUEST model, while Figure 4(b) shows a tangent. In the tangent diagram, note the lack of the Consequence (C) arc between State 2 and Event 5, and the lack of an Initiates (I) arc between State 2 and Goal 3. On average, stories created in the whiteboard interface had 0.6 backchains, with each backchain being a tangent. Stories created using the computerized interface averaged 2.67 backchains, 1.67 of which were also tangents. Note that the standard deviation (see 1) exceeding the average is an effect of some authors not using any backchains and the impossibility of having a negative number of backchains.

**Anecdotal Evidence**

**Past Experience**

Throughout both groups, participants reported little or no experience with writing or telling stories. Typical writing experience included keeping a journal or working with a literary magazine in high school. One whiteboard subject described using a simple storyboard to tell stories with her children. Subjects reported oral storytelling including telling children bedtime stories or sharing anecdotes with friends. Several participants in both groups also said they did not tell stories, feeling that they were not good at doing so.

**Whiteboard Interface**

Whiteboard users’ comments about the interface frequently dealt with the nature of using a tangible interface rather than a computer. Some commented on the static nature of the Post-It notes, feeling unsure that they could make changes to events they had already written:

Interviewer: Was there anything we could have changed about this tool to make it easier?

Subject: No, I don’t think so. Well, I mean, it’s Post-It notes, so it’s hard to edit them or really . . . Once they’re down, they’re down, so obviously, if they were some sort of digital artifact that you could overwrite, that might make things easier if you want to revise things.

Concerning revision, another participant commented on the difference between the static whiteboard interface and a potential electronic interface, noting she might have revised in the latter case but did not know if she could on the whiteboard. Despite these apparent issues with the Post-It notes,
other users did comment on the ease with which they could move them around the whiteboard, whether to re-order them or to give themselves more space to work.

**Computer Interface**

Participants in the computer interface condition specifically noted several issues with the user interface. Most importantly, some users felt that the arrows were pointing in the wrong direction. As users created backchains, the path of the arrows tended to be the reverse of temporal action and, therefore, the reverse of causality as well. In QUEST models, arrows indicating causal relationships point forward temporally, but arrows indicating character goal relationships can sometimes point in the opposite direction of temporal flow. Thus, the users’ own tendencies toward revision were confusing their interpretation of the arrows in the story graph.

Other common issues dealt specifically with the design of the user interface. One frequently reported issue dealt with the creation of new events. Each new event added by a user appeared in the same location near the upper left corner of the canvas. Users reported disliking having to drag every event and, as their stories grew, scroll back to the top of the canvas to retrieve the newest event. One user reported that this trait led him to write longer events to avoid scrolling back. Users were inconsistent, however, in describing alternative methods of placing new events. Users suggested being able to click the canvas to place a new event, or having the UI search for an empty space, possibly near other recently created events. Several subjects also reported running out of space, leading to their screen becoming cluttered. One participant noted that he thought that the canvas would grow exponentially, but he admitted to not knowing what led him to this assumption.

**Algorithm**

Over the course of the interview, subjects commented on whether they felt compelled to answer questions. One computer interface subject reported that his personality led him to answer the questions:

Subject: There were a couple of times where I felt like [the questions] kind of got me sidetracked, but I’m kind of anal about some things and I didn’t really want to leave a question unanswered.

Another subject in the computer interface condition said that while he was not compelled, he did not feel that there was much else to do, because of the simplicity of the story. He did recognize, though, that the questions were an opportunity to revise his story. Subjects also reported being distracted by the questions, and noted that they could distract from the story writing process. One computer interface subject noted that the questions “were a bit bugging at times, because...it seemed that they were kind of interrupting the flow of the story...” However, several users in both conditions did report the questions useful, encouraging them to fill in details when they had been unintentionally vague.

Users in both groups seemed to be acting to make the questions go away. Said one computer participant, “I basically was trying to come up with a way to author an event that would prevent you from asking me more questions down there.” A whiteboard participant also reported that he was thinking that the “quickest way to finish a story in the program, perhaps, is to satisfy all of its questions.” Therefore he sought to write events that could be unambiguously interpreted and hopefully avoid a question. Similarly, subjects in both cases reported anticipating questions, and therefore attempting to pre-emptively respond to questions by giving additional detail in new events. These participants seemed to be playing a game, hoping to do something that would make the system stop asking questions. This might indicate that they answered these questions despite not feeling that doing so was worthwhile or would help improve the story. Thus, users were being pushed to make revisions that they felt were unnecessary.

**Personal Experience with ReQUEST**

Because of the nature of the Wizard-of-Oz study format, several subjects believed that ReQUEST, particularly the computer interface, would work well in a collaborative environment. Some subjects compared the back and forth aspect between author and system (or Wizard) to a chat room or instant message. One computer subject felt it was important that the Wizard was writing the questions, rather than a computer:

Subject: I like the aspect of the program where you’re able to interact with someone...The way you can add to the conversation at the same time is good.

Interviewer: Would you feel that way if it were just the computer asking questions? If I weren’t here, but the computer still asked questions, do you think you’d feel the same way about it?
Subject: I think it would seem less useful...But only because of the nature of someone being there makes the experience more significant.

This type of statement indicates an unexpected influence of the Wizard. Authors may have been more willing to revise because they thought a human reader might not understand rather than a person simulating the actions of an artificial intelligence.

DISCUSSION
Because participants using the computer interface answered a larger proportion of questions and created more backchains and tangents than those using the whiteboard interface, we believe that the computerized interface encourages story revision. By revision, we are referring to adding events or state information to clarify or elaborate on certain aspects of the plot. We believe that these revisions were intended to provide readers with more detail, particularly since users in the computer interface condition answered more questions, and frequently went backward in the timeline of the story in giving these answers. Because writers were free to ignore questions, they could choose to leave out details that they considered unimportant to the story. Additionally, writers could have chosen to answer questions with existing events, thus not revising their story, but indicating causal links within the portion that had already been written.

The notion of revision discussed above corresponds to Sharples’ [15] model of writing as creative design. In this model, human creative writing behavior is a cycle of engagement and reflection. In the Engagement phase, the human author actively generates new concepts by querying memory for relevant patterns in a forward-chaining fashion. In Reflection, the author ceases new concept generation and analyzes his work before elaborating on what has already been written. ReQUEST, by asking questions, encourages reflection. In the computer interface condition, questions served to encourage readers to spend more time in Sharples’ reflection phase, as compared to the same AI behind the whiteboard interface. As previously noted, computer users answered a higher percentage of their questions, frequently going back in the story to do so, which in turn led to backchains and tangents. These actions indicate that authors in this group were taking time to stop writing, think about their story, and make improvements based on the feedback given by ReQUEST.

Comparison of Stories
Given that subjects in the computer condition spent more time in the reflection phase of Sharples’ [15] model of creative writing, we wanted to know whether this action had an effect on the quality of stories. For this reason, the eleven stories were given to four judges to compare. Each judge was asked to do a round-robin comparison of each whiteboard-created story to each computer-created story, resulting in 30 total comparisons. Each judge was given a randomized list of the 30 pairs of stories and told to pick the better story from each pair. Half of the combinations listed a whiteboard story first, with the other half listing a computerized story first. No criteria were given for choosing the better story, except that the focus should be on plot rather than on natural language aspects. Judges were told to read stories in the order they were listed for each pair. Judges were also told to re-read stories, even though they may have already read them earlier in their comparisons.

Stories were given to the judges as an ordered list of events. Story events were placed in temporal order as could best be ascertained. In the case of stories from the whiteboard interface, we used the timeline created by participants with the Post-It notes on the board. However, with stories from the computerized interface, when the temporal order of events was unclear, we followed the heuristic of placing state descriptions at the beginning of the story, and placing events as late in the story as would make sense. Additionally, to avoid biasing factors, some minor elements of stories were modified before being given to judges. These modifications chiefly consisted of language touch-ups to provide consistent quality of language. In some instances, stories had an event removed that duplicated much of the text of an immediately preceding or following event. One story had two events combined to resolve the fact that one was a sentence fragment. Also, one story originally used letters to identify characters; these letters were replaced with names in hopes of avoiding bias against unnamed characters.

In general all four judges preferred stories created in the whiteboard interface over those from the computer interface. Overall, whiteboard stories were favored in 76 of 120 comparisons (63.3%). Each story in the whiteboard group won at least half of its comparisons, with one story from the whiteboard condition (Story E) winning 22 of 24 comparisons (91.7%). Only one of the six stories from the computer interface condition won more than half of its comparisons. Even when removing Story E, stories from the whiteboard group were still favored by a margin of 54 to 42 (56.25%).

Too Much Revision?
However, the fact that stories from the whiteboard interface condition were consistently rated better than those in the computer interface condition suggests that the latter may have gone beyond encouraging reflection, to the point of convincing writers that revision was necessary when, in fact, doing so would be to the story’s detriment. While a good tool should encourage writers to reflect on their content, we believe our data indicates that excessive reflection is harmful to the product and does not help in creating work that might be valued by the community. This possibility of unneeded reflection is confirmed by interviews with subjects, who reported feeling compelled to answer questions despite knowing that they had been told that they could ignore questions indefinitely. Subjects also reported feeling that the questions frustrated them or distracted them from the main story, but answered them nevertheless. Csikszentmihalyi [1] has argued that creativity results from a human having an uninterrupted flow of concepts. These comments from our participants, as well as their actions, seem to indicate that they lost this uninterrupted flow, and revised their story beyond the
This study lacks a base case of subjects telling a story without an algorithm prompting them with questions. With this group, we would be able to say with greater confidence that our conclusions concerning revision were a result of the interface alone.

All subjects in our pool came from a technical background, with most working in computer science labs throughout our building. As a result, our pool was not especially diverse, potentially leading to biased results. However, we believe their technical proficiency did not allow them to use our relatively simple interface any better than a less technically experienced novice writer would.

Participants in our study reported a certain level of “performance anxiety” while telling their stories. Subjects noted that the presence of the Wizard in the room constituted a human audience, who could be looked to for simple responses such as smiling or laughing, as well as noting when the Wizard was writing notes or questions. On a related note, because of the one-on-one nature of the study, some users found the process to be good for collaboration; computer interface participants compared the experience to using a chat room or an instant message. However, when asked, subjects reported that they did not believe their stories would have changed significantly if there had been nobody else in the room.

**Future Work**

In the future, the ReQUEST interface and ReQUEST algorithm should be updated to take user feedback into account. Such an update could consist of minor changes to the existing computer interface, implementation of the whiteboard interface, or the development of a new interface. Any future interface should account for the inclination toward revision that was apparent in this study, as well as addressing other issues raised by users during interviews. In addition, we would like to add artificial intelligence to the ReQUEST interface, allowing the QUEST model and questions to be managed by the system rather than a human acting in its place. With these changes, ReQUEST could further assist novice writers in creating stories for consumption.

Ultimately, ReQUEST could be most effective in combination with other tools for authoring in specific media. As it stands, a story written in ReQUEST could be applied to any medium of the author’s choice. However, ReQUEST could be added to authoring systems for specific media, such as screenwriting or machinima. Authors would then have the opportunity to write the story with the assistance of the ReQUEST algorithm, then deal with other aspects of the story, perhaps visual, through the partner system. Such work has been discussed in [12], but the extent to which an assistive authoring system can be combined with other media remains an open question.

**RELATED WORK**

Advance planning and structuring is common in almost every form of writing. Plot structure organization is particularly common in screenwriting. The Save The Cat™ method

<table>
<thead>
<tr>
<th>Questions Generated</th>
<th>Computer Interface Without S7</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of Questions Answered</td>
<td>67.4%</td>
<td>24.7%</td>
<td></td>
</tr>
<tr>
<td>Questions Answered With New Events</td>
<td>78.0%</td>
<td>28.0%</td>
<td></td>
</tr>
<tr>
<td>Percentage of Events Created Out Of Temporal Order</td>
<td>30.4%</td>
<td>21.1%</td>
<td></td>
</tr>
<tr>
<td>Percentage of Events Used To Answer Questions</td>
<td>61.5%</td>
<td>35.5%</td>
<td></td>
</tr>
<tr>
<td>Backchains</td>
<td>1.4</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>Tangents</td>
<td>0.6</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Data with outlier S7 removed
provides novice screenwriters with fifteen “beats,” or specific themes, and goes as far as to specify on which pages of the screenplay these beats should appear. Additionally, other commercial tools such as Dramatica™, Power Structure™, and Truby’s Blockbuster™, advise authors with organizational methods and plot structures based on common screenwriting techniques. While these tools rely on typical screenwriting practices, we aim to differ by not directing writers, of any variety, to any particular approach or style.

Several interfaces have been designed for providing feedback to writers. Most narrative authoring systems developed thus far are for interactive narrative environments [8][9][17]. However, other systems have been designed explicitly to aid children and students trying to tell stories that are less complicated than those typically associated with interactive narrative. StoryStation can provide assistance for children trying to tell a story, but its intelligence works best when users are re-telling a story they have already heard [14]. Select-a-Kibitzer provides scaffolding on student compositions from numerous agents, including advice that might conflict with other agents’ suggestions [18]. Agent feedback may take the form of requesting clarification between two sentences or pointing out spelling errors, among other possibilities. ReQUEST is meant to provide assistance to all novice writers, not just students trying to understand stories or learning to write. Because ReQUEST focuses primarily on events and actions, the algorithm does not perform any checks on spelling or conceptual transitions in the work. ReQUEST authors enter the main ideas of actions rather than working out the full text of their story.

ReQUEST shares similarities with narrative understanding systems, such as [5][2][19][11][10]. Like ReQUEST, AQUA [11] attempts to understand narratives, modeling them and using questions to resolve anomalies in the narrative. In ReQUEST, anomalies indicate character goals lacking motivation and events that do not fit into the causality of the narrative. However, because ReQUEST is modeling the story-so-far in the midst of authoring, it is prone to finding anomalies that exist only because their resolutions have yet to be written. Authors answer ReQUEST questions to explain the anomalies in their narrative.

Sharples characterized writing as design, and proposed that the writing process consists of a cycle of engagement and reflection [15]. The actual production of text takes place within the engagement phase, while the reflection phase includes review and organization of the content. ReQUEST offers users the opportunity to engage by spelling out the actions of the story, while reading, considering, and answering questions posed by the system makes up reflection.

Like StoryStation and Select-a-Kibitzer, Adventure Author was also directed toward children [13]. Adventure Author works from the same perspective as Sharples, that creating a game, just like telling a story, is an example of creative design. Students who used Adventure Author entered the reflection phase as they watched their peers play their games and struggle. Seeing their difficulties encouraged them to go back and alter their game. Such actions have clear parallels in storytelling.

CONCLUSIONS
We have described a comparison between two user interfaces for an intelligent story authoring assistant, ReQUEST. In this study, users of the computerized graph-like interface revised their work more frequently than those who used the whiteboard interface. Further, subjects in the computer condition may have had their flow interrupted, leading to a loss of creativity. This interruption may have led them to reflect and revise more, possibly to the detriment of their story. These revisions frequently resulted in backchains and tangents in their stories. These stories also saw a larger percentage of questions answered, with some subjects feeling compelled to answer the questions posed by the system. Finally, stories created in the computer interface were consistently judged to be worse than those created using the whiteboard. While it is good to see novice authors reflecting on their work, the apparent need to answer questions, despite being aware of going on tangents and possibly harming the story, is a cause for caution when building intelligent creativity support systems.

The computer-as-audience approach merges several of the traits of the computer-as-coach and computer-as-colleague approaches to creativity support, but care must be taken to preserve productive creative flow. In general, there is a growing demand for a means of creation of media content. While tools exist for manipulating data that can be used to produce almost any conceivable form of media, the goal of supporting non-experts in the creation of valuable content remains an important goal.

ACKNOWLEDGEMENTS
Special thanks to Peter Danenberg, Jonathan Rowe, Julia Kim, Stacy Marsella, and members of the Georgia Tech Intelligent Narrative Computing group.

REFERENCES


