NOVICE-ORIENTED AUTHORING INTERFACES FOR PLAN-DRIVEN INTERACTIVE STORIES

JAMES SKORUPSKI

ABSTRACT

The authoring of interactive, generative narrative is a task that typically requires an extensive multi-disciplinary background in computational and narrative theory. With the increased popularity of large story-driven interactive games, the demand for truly dynamic narrative systems has grown, while the number of cross-disciplinary experts that can both design compelling plots and utilize the underlying computational power has stayed relatively small. This work aims to develop a series of complete, publicly releasable authoring interfaces that allow novice, non-technical writers and game designers to easily author plan-based interactive stories, and thereby more closely connect these creative minds to concrete implementation of their ideas. Through a process of iterative development driven by constant evaluation and lessons learned from each subsequent system, we hope to develop a series of truly usable, writer-friendly, and powerful interactive story authoring environments, and at the same time gather valuable and widely applicable insight into the requirements of interfaces aimed at novice-oriented authoring.

INTRODUCTION

The creation of high quality interactive stories requires both expertise in authoring (compelling plot arcs, dialog, character conflicts, etc.) and technical expertise in computational and generative story representations. Currently, authors, or more typically author teams, must have competence in programming, a background in artificial intelligence, competence in interaction and game design, and skill in story authoring, including the creation of compelling plot spaces (potential plots), rich characters, and believable dialog. Even for experts, the necessity of bringing all these skills to bear can make the authoring of complete interactive story experiences a heroic undertaking; for novices, the creation of such experiences is out of the question. The interactive drama Façade [MAT03], for example, uses over 200,000 lines of reactive behavior planning code to represent a 30-minute dynamic player experience. These great demands on authors relegate interactive story authoring to a small group of experts, and limit the exploration of the potential design space of interactive narrative. Furthermore, authoring complexities hinder widespread adoption of these technologies in the game industry, which stands to benefit from games with meaningful social interactions and dynamic story spaces that are much larger than those that could be realistically hand-authored.

These daunting challenges motivate the development of practical and usable authoring interfaces that can harness the computational power of generative interactive story models and still appeal to *novice* users, or those with little to no programming experience. These types of users include writers and game designers, who traditionally have limited interaction with the underlying computational framework powering modern interactive experiences, and are traditionally required to translate their story concepts into a common language understood by technical experts. This work aims to make significant progress in eliminating this layer of translation, allowing creative minds to have a more direct connection to the implementation of their vision. In short, we wish for writers and designers to focus on writing and designing, not programming and debugging.

RESEARCH QUESTION

How can novice authors create interactive stories with wide variability?

The above question guides the progress of our research and the multiple iterations of its evolution. In order to address the query, this dissertation work aims to develop, in three primary iterative stages (as seen in Figure 1), complete authoring interfaces that allow these users to design high quality interactive experiences without ever being required to look at or touch a line of code. For each stage, we pair a chosen story generation model with an interface built with our target audience in mind, and put it in the hands of many real novice authors that create interactive stories. Using the feedback from each round of evaluation, we hone the interfaces, addressing the weakness and accentuating the strengths of each tool. The lessons learned from each stage inform the development of the next, resulting in an exploratory process that allows us to identify the primary challenges and requirements in creating practical and novice-friendly story authoring systems.

We have already developed the first two of these stages, named Wide Ruled and Story Canvas. These systems combine a version of an established hierarchical story planner, called UNIVERSE, and with traditional windowed or graphical storyboard-based interface, respectively [SKO07, SKO09, SKO10]. From a complete evaluation of the Wide Ruled authoring tool, we discovered and analyzed the challenges of interactive story creation and used these lessons learned to inform the visual authoring metaphors and author assistance present in Story Canvas. The next research step is to evaluate the storyboard-based interface, and gain insight into novice authoring challenges experienced with the newer visual interface. The last stage of our work will take the evaluated and refined visual storyboard metaphors of Story Canvas, and apply them to a different underlying story generation model, driven by a previously developed "social physics" engine built on the concept of characters that participate in social games that dynamically alter the state of the story world [MCC10]. In addition to providing a different computational story abstraction, this new story generation core will also provide a domain for an active planning system that will constantly provide dynamic story event suggestions that can ensure the progression of an overarching plot arc, or maintain a constant pace of story world evolution during the execution of the resulting experience. This new "mixed-initiative" authoring interface, using the established visual elements of the previous stage of research, will play a more active role in authoring interactive stories, providing inspiration and guidance for novice users that wish to harness all of the capabilities of a complex underlying story generation model. It will be built to encourage a continuous, iterative, and creative exploration between author and interface.

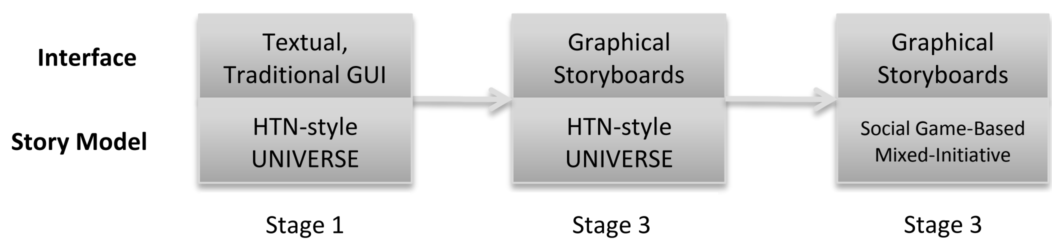

Figure 1. The stages of our research

CONTRIBUTION

The contribution of this work will be a set of complete, publically available authoring interfaces that provide a novel method of interaction with various underlying story generation models, as well as the practical design knowledge gained by our evaluations of these systems with novices. We hope that each stage of our work will improve upon the last, and that the interface designs and methodologies we develop during this iterative improvement will provide a wider applicability to other novice-oriented computational narrative systems in academia or industry. No single story generator is ideal for modeling all interactive narrative experiences, but if our work can improve the ease of use and approachability of other authoring interfaces for different story models by informing their design process, than we will have succeeded.

RELATED WORK

The work in this dissertation is based on and related to many areas of research that cover both the underlying computational model of each system, as well as the methods of interaction and goals for each interface. The story generation and authoring community has provided us with a long history of story representations and generation styles, while the authoring community has spanned the gamut from static to dynamic story creation, for both domain experts and novices alike. The basis of Story Canvas and our third stage of research, storyboards and comics, is grounded in existing detailed analysis in the fine arts, and many of the challenges that our work faces introducing non-technical users to computational concepts have been researched for many decades in the software engineering field.

Story Generation

There are many existing examples of story generation models, including classic systems such as UNIVERSE, Tale-Spin, and Minstrel [LEB84, LEB85, MEE81, TUR94] that are designed around primarily author-level or character-level concerns, as well as more modern hybrid approaches [SIM05, WEI05] . For our Wide Ruled and Story Canvas interfaces, we chose the UNIVERSE model of story generation, developed by Michael Lebowitz in 1984, that demonstrates an early HTN-like model of plan execution. As a classic story generator, UNIVERSE was the first system to model the pursuit of authorial plans, which coordinate multiple characters, in pursuit of authorial goals. This style of story execution was selected for this research because its success with students in previous Interactive Narrative classes, who preferred the top-down goal model as an easier and more intuitive way of constructing stories [SKO07]. While this model of generation is simple, it was never originally designed to model interactive stories. As a result, we implemented two different forms of story interactivity, based on planner intervention and asynchronous goals, throughout the evolution of Wide Ruled, and selected the asynchronous method with refinements for the development of Story Canvas. More details on these two methods of interactivity appear in later sections.

For the third stage of our work, we chose a character-driven model of social interaction, called *Comme il Faut* (CiF), developed by McCoy et al [MCC10]. *CiF* is a computational model of social games in which agents in a virtual world participate. Each of these individual games are short dynamic interactions between two or more characters acting in specific roles, complete with templated dialogue instantiation and detailed internal social effects on the internal state of characters as well as relationships between characters throughout the simulation. To demonstrate the effectiveness of this model, McCoy describes the implementation of an interactive game called "The Prom", which uses *CiF* and its social games as core interaction mechanic in an exaggerated high-drama environment of teenagers in high school. Our third stage of research, as described in detail in a later section, will be utilizing some of the vast amount of pre-authored social games and game-related knowledge in The Prom as a way to provide a basis for a rich domain on which our mixed-initiative authoring interface will base its recommendations.

Novice-Oriented Story Authoring

The complexities of authoring interactive stories have been discussed in detail in previous work [PIZ08, SPI09]. Many modern novice-oriented story authoring tools are built around the creation of story graphs, which require an author to statically represent every potential story path [BAL07, BER02, GOB07, WEI05, ZAG06]. While these story graph systems utilize a readily understandable and visualizable model of the space of potential stories, they lack the power of a generative formalism such as our own model, and require the explicit creation of a combinatoric space of story fragments, ultimately limiting the size and variability of the interactive story space to that which can be reasonably specified manually by the author. Because of their relatively simple computational model, story graph-based authoring systems are among the most popular novice-oriented authoring systems. In order to keep a familiar structure but add computational power, some authoring systems have employed a hybrid system of story branches and planning [CHE08] or story branches and rule triggering [IUR06] to modify the story tree.

For the last stage of our work, we aim to build an authoring system that collaborates with the author to create an interesting interactive story. We will use a high level story planner and a planning domain metatheory based on the *CiF* social games model. The concept of using AI techniques, specifically planning, to facilitate human-machine collaboration in a creation is not new, and is referred to as *mixed-initiative* interaction [BUR96]. The application of mixed-initiative planning to the domain of interactive stories is a more recent development [THO06]. In the realm of mixed-initiative authoring applied to the domain of interactive stories, there exists some more recent discussion of the design requirements and potential authoring experiences of a conceptual system for constructing individual behavior plans in an interactive narrative environment [KRI07, KRI072]. While this speculative work provides some guidance in the important components of a novice-friendly narrative authoring environment, it does not detail the complete design of any concrete interface of the type described, and focuses on a less structured character-centric emergent narrative experience, which contrasts with our desire to maintain author control over high level plot progression. Additionally, there has been recent work in graphical plan domain authoring with *Bowyer*, an extension of the *Bowman* HTN-based mixed initiative narrative authoring system [CAS09, THO062]. *Bowyer* is graphical interface used to simplify the process of creating and integrating declarative planning domains into the procedural framework of a modern virtual world infrastructure, in order to construct complete interactive experiences. This work focuses on aiding planning researchers in generating, managing, and executing highly technical plan domain specifications. Our work, in contrast, focuses on a providing a novice-focused interactive narrative authoring environment, with an authoring interface that closely mirrors the resulting interactive experience.

Storyboards and Comics

The sequential art of storyboards have been an extremely effective spatio-temporal visualization technique for films, comics, graphic novels, computer animation, and game design, and there has been extensive previous analysis of the visual techniques and metaphors that are present in the medium [EIS85, MCC93]. This history and analysis provided a rich vocabulary of visual techniques, such as line styling, text arrangement, panel composition, shadowing, and color choice, for conveying a range of emotions and messages. It is an established background for the design of our Story Canvas authoring system, and inspired the visual metaphors used to interact with the underlying model.

Comic-Based Story Generation and Authoring

Storyboards and comics are not a new interface metaphor, and have previously been used to automatically visualize generated story plans from domains authored in a non-visual, expertfocused fashion [PIZ082]. While these systems generate storyboards from generated plans in a domain, we aim to address the inverse problem – creating plan domains from storyboards.

Gingold's *Comic Book Dollhouse* is an authoring system used comic strips to represent visual interactive stories, but did not generate multiple story variations and depended on users dropping graphical icons on story panes to trigger various story paths, akin to a the navigation of a static story graph [GIN03]. *Longboard* is a system that used author-created storyboard sketches to guide a planner used to generate camera shots in a virtual 3D world [JHA08]. While this storyboard interface does not allow the user to author the behavior, plot progression, or interactive features of the story, it provides us with a critical step in the right direction.

Recently, there has been work on an authoring tool, called *ENIGMA*, that lets novices author linear non-interactive storyboards with an interface that suggests potential actions for the selected characters and objects in each scene [KRI10]. These storyboards are single instances of annotated events in a story, using an established collection of characters, actions, and effects. The suggestion information presented to the author is based on knowledge trained by previous novice linear storyboard authors using the same authoring tool and creating similar annotated storyboards. In contrast, our storyboard systems target the authoring of non-linear interactive, generative stories, and our interface will base its suggestions dynamic knowledge of existing story structure and a social game model that is not specific to any pre-authored characters, events, or states in the story world, ultimately collaboratively creating an authoring environment unique to the experience being authored.

Visual Programming

With the move to the visual domain, Story Canvas and the final stage of our research begin to meet the same sort of challenges that visual programming languages must encounter: introducing potentially complex computational constructs in a simple and non-cluttered way for potentially novice users. While general visual programming has traditionally been considered arguably ineffective and inefficient for large and complex systems due to scale (WHI97), domain-specific visual languages have been successful in many practical instances. These include the sound processing and music synthesis tools such *Max/MSP* and *Pure Data*, and graphical user interface builders in programs like *Eclipse and Visual Basic/C++* . Our storyboard authoring work is closely related to a domain-specific visual language, because we are restricted to the computational constructs and visual metaphors of our chosen story models. For this reason, we believe that our limited domain of visual-to-conceptual mappings will allow us to avoid the pitfalls of general visual programming.

Closer to our domain of interactive media, systems such as *Storytelling Alice* and *Game Maker* have successfully provided visually enriched game authoring experiences, but require some programming of textual scripts to create complete games [KEL07, YOY10]. *Kodu*, introduced in 2009, is a successful and completely visual game programming environment for young children, but is limited to a uniform model of physical movement and interaction, with very limited support for the abstract computational structures and alternative non-physical interactions desired in interactive narratives [MAC09].

In the domain of comic-based programming, previous research by Fernaueus and Kindborg et al discusses graphical rewrite rules, or graphical pattern matching and modification, as well as more complex general programming constructs represented in comic form [FER06, KIN05, KIN07]. While this work focuses exclusively on the limited domain of purely graphical modifications of visual objects, our system combines the explicit visualization of story objects with visual representations of abstract logic and computational constructs specific to the domain of interactive stories.

PREVIOUS WORK

WIDE RULED

The first stage of our research is the Wide Ruled text-based story authoring tool, which uses a traditional windowed user interface tailored to novice users, combined with a UNIVERSE-based underlying story model for generation.

STORY MODEL

As previously mentioned, the story generation model in Wide Ruled and Story Canvas is based on the HTN-style UNIVERSE model of story planning which models story structure as a set of hierarchical plans that encompass one or more ways to accomplish a story goal for the author. A Wide Ruled story contains a set of author-created story objects, represented as *Characters* or *Environments*, each with associated attribute-value trait pairs, and relationships to other characters or environments, respectively. The story also contains a set of *Plot Point Types*, which define categories of episodic attribute-value data generated and utilized only during the story generation process. *Author Goals*, with optional parameter variable inputs, are the primary unit of story planning in this tool, each one containing one or more *Plot Fragments* that describes a set of actions that fulfill its parent author goal. Plot fragments have ordered *Precondition constraints* that must all be true before execution. These constraints rely on the current state of the story world during generation and can also bind story objects and their attributes to local variables for later use in that same plot fragment. Additionally, plot fragments contain a list of sequential *Story Actions* that can modify the story world during execution. These actions can modify story characters, environments, and plot point instances bound in the precondition, create and delete instances of plot point types, calculate new values, print out story text parameterized by bound variables (the actual textual output of the story seen by a reader), and pursue other author goals. A complete description of story actions is detailed in our previous work [SKO07].

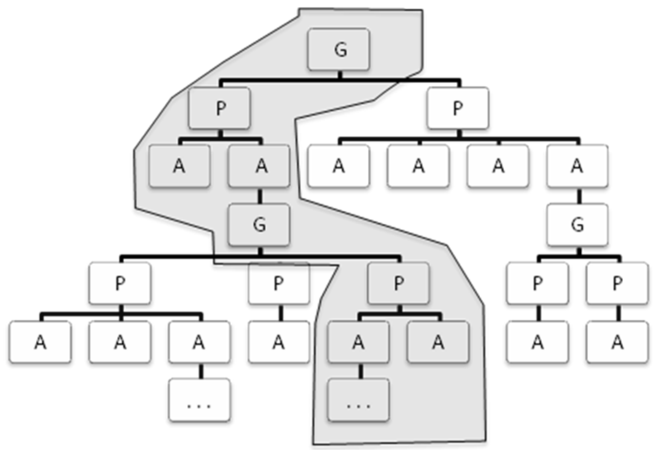

Figure 2. The Wide Ruled/Story Canvas Story Hierarchy. "G" nodes represent author goals, "P" nodes represent possibly selected plot fragments, "A" nodes represent sequentially-executed story actions, and the gray overlay represents a single instance of a story, executed top-down, within this potential story space.

Generation

Story generation in Wide Ruled begins with a top-level initial author goal, which randomly selects amongst all executable plot fragments with valid preconditions, and then sequentially executes all of its contained story actions to successfully complete a story. If the generator encounters a story action that pursues another author goal, the process repeats and a new random plot fragment is executed. The relationship between author goals and their associated plot fragments describes a potential tree-like space of stories, in which a single generated story is represented as a traversal of the tree, as seen in figure 2. The original UNIVERSE model was not interactive, so we implemented story interactivity in two different ways for each version of Wide Ruled, one method based on plot fragment selection intervention, and the other based on asynchronous goal execution. Both of these interactivity models are described in the later sections of this paper.

Output

The output of a Wide Ruled story is a string of sequential text generated by any plot fragments that display text as part of their story action list. The variation in each textual story derives from the parameterized nature of each of these blocks of outputted text. Information from characters, environments, and plot points in the story world are captured and stored within preconditions inside each plot fragment, and then modified and printed to the screen in these text blocks. A combination of static hand-written text, interspersed with that dynamically bound variable information, results in story variations dependent on the particular story being generated. This resulting text is displayed in a reading interface that prints the text in real time as it is generated by the system.

Wide Ruled 1

Wide Ruled 1, developed and evaluated in 2007, provided a familiar graphical interface to the underlying UNIVERSE-style story planner, utilizing standard interface conventions, including

OK/Cancel window actions, editable tables of attribute-value pairs, clickable item lists, and hierarchical and collapsible tree lists. In addition, the interface contained story-centric terminology for each component of the interface, avoiding technical terms where possible to avoid confusion for those without a technical background. This became troublesome when attempting to describe the binding and ordering of variables, passing of parameters, and complex precondition constraints, which are inherently technical in nature. A complete documentation of all the features in Wide Ruled 1 is described in detail in our earlier work [SKO08].

Interface Components

Figure 3 shows three of the key interface components of Wide Ruled. The main screen displays to the user the lists ofstory objects (characters, environments, and plot point types) alongside the structure of the story (the author goal-plot fragment hierarchy) on the right side. The three story object lists can all be edited, and the modification of any single story object is similar to the character editor displayed in the middle portion of figure 3. The story structure list in the main interface represents the story hierarchy as an intended hierarchical list, with each element of that list editable. Author goal editing is similar to the editing of any story object, except that goals contain only a list of parameters to be passed into author goals. The plot fragment editing interface is the most complex part of the Wide Ruled authoring process, and thus was the focus of our efforts in adapting our story generation model to novice users. A screenshot of this interface is at the bottom of figure 3.

Plot Fragments

Plot fragment editing requires the author to focus on two major components: the precondition, and the list of actions. During generation of a story in Wide Ruled (and Story Canvas), the precondition for each plot fragment is a list of requirements that determines whether it is eligible to be selected as a possible way to complete an author goal. These requirements are a list of constraints on the traits and relationships within characters, environments, and plot points, and every constraint must be true simultaneously for the plot fragment to be valid and ready to use within a story. In addition to providing an eligibility test, preconditions bind information from characters, environments, and plot points to local variables for use and modification within the plot fragment. In order to simplify the understanding of complex precondition constraints and story actions, Wide Ruled 1 and 2 displayed these elements as natural-language statements. For example, consider the following constraint in a plot fragment precondition is typical of a story in our tool:

```
There exists a character, saved as “myChar”, where trait “alive” = true, and
relationship “enemy” target name is saved as variable “enemyName”, and trait
“name” is saved as variable “charName”.
```

In this case, any character that is matched must have a value of “`true`” for trait “`alive`”, the name of the character is bound to the variable “`charName`”, and the name of the enemy of that character is saved to the variable “`enemyName`”. In addition, a reference to the matched character, “`myChar`” is also saved for later editing in the plot fragment’s story actions.

As mentioned in an earlier section, story actions are sequential tasks that are performed when all of the constraints of a precondition are satisfied and the plot fragment in which they are contained is selected for execution. These actions modify story objects, calculate new variable values, create or delete plot points for the storage of temporary story information, print parameterized story text, and subgoal other author goals. The following text is an example of a story action described in natural language, as seen in the Wide Ruled interface, that modifies the "`name`" trait of the bound character "`myChar`" from the previous precondition constraint example:

```
Edit Character "myChar": set trait "name" to "John"
```

These descriptions help users to quickly understand what a plot fragment does without having to decode cryptic symbols. Due to the complexity of creating precondition constraints, Wide Ruled 1 provided a wizard-based interface to create these potentially complex statements [SKO07]. Similarly, wizards were used to generate story actions, which often required referencing a bound variable for modification or printing. For a more detailed analysis of plot fragment editing, the full range of precondition constraints and actions can be seen in our previous work [SKO07, SKO09].

Interactivity: Planner Intervention

As mentioned in the Related Work section, we implemented a story interactivity model on top of the traditionally non-interactive UNIVERSE-style generator. The reader could intervene in the story planning process by selecting among possible plot fragments for some author goals. Authors could select a single character as the "active character" in each plot fragment, and a reader would select a character when reading, and select amongst the possible plot fragments available for that particular character, if any were available. This method of interaction proved troublesome and non-intuitive because it attempted to shoehorn the concept of an "active character" for a decidedly non-character-centric story planner [SKO7].

Wide Ruled 2

Wide Ruled 2, like its predecessor, uses a traditional windows GUI and text-based output, but addresses many of the shortcomings of Wide Ruled 1, which were made apparent with the feedback received from the initial user evaluations in that work, as well as through the continued use of the tool by its authors during development. It is built off of the same graphical interface style of the original version with a series of improvements described below, and is the publically available version that is currently used in classroom settings and is actively supported. Wide Ruled uses the same story model and generation algorithm as the previous version, described in the previous section. Figure 3 shows the main screen of Wide Ruled 2. For a detailed description of the Wide Ruled 2 see our previous work [SKO09].

Interactivity: Asynchronous Goals

The underlying story generation model of Wide Ruled was modified in this latest version in order to provide a more intuitive authoring and reading experience. The interactivity model of the previous iteration proved to be unnatural, as described above. To address this, Wide Ruled 2 implements an asynchronous goal execution model of interaction, in which story authors specify a set of *Interactive Actions* for each plot fragment, which can be executed at any time during the execution of that fragment. These actions are separate goals, which can modify the story world and output completely new panes to the screen. During generation, when an interactive action is activated, the current fragment is halted, and the interactive action is executed completely, allowing for a dynamically interleaved story with full authorial control over what a reader can do at different times during generation. This change to the interactivity model also required that the story execution be slowed down so that a reader could choose to activate these actions before the story was completed.

Changes from Version 1

Wide Ruled 2 introduces a new underlying story generator that is driven by the ABL reactive planning architecture used in interactive dramas like Façade [MAT02]. This planner utilizes a similar hierarchical decomposition as that of the UNIVERSE model, and, due to its reactive nature, easily facilitated our new interactivity model. The new reader-driven interactivity model also prompted another change to the underlying story model. In Wide Ruled 1, if an author goal had no executable plot fragment, the generation loop would back-track its execution, erase any outputted text, and choose another valid plot fragment in a previous author goal in the execution stack. This model was not ideal for the online, read-as-you-execute text output model of story generation that is required of the new real-time interactivity model, and would result in the removal of text that was already viewed by a reader. As a result, backtracking was removed from Wide Ruled 2; if the story generator encounters an author goal with no valid plot fragments, story execution halts.

The wizard interfaces used in Wide Ruled 1 to create new precondition constraints and story actions were removed in version 2 in response to user feedback. The slow nature of the step-by-step guidance through the creation process became cumbersome once a user became proficient in plot fragment editing. Wide Ruled 2 implements a more direct, list-based interface that allows the user to edit all components of a constraint simultaneously. The initial learning curve required to use this new interface proved to be minimal and reduced authoring time for most authors [SKO09]. Similarly, the wizards used to create and modify story actions were simplified into one-screen editors, with limited wizard-like steps.

Evaluation

Since 2007, Wide Ruled 2 was used in three different classroom settings by a total of 91 students with mixed non-technical (digital arts, new media, literature) and technical (computer science, game design) backgrounds. At the University of California, Santa Cruz, we included Wide Ruled 2, like Wide Ruled 1, in two sections of the Interactive Storytelling class hosted by the computer science department. This class is cross-listed as an undergraduate and graduate computer science and digital arts and new media graduate class, allowing a varied (although a majority technical) audience to learn the theory, techniques and technology behind the creation of interactive stories. The third classroom setting for Wide Ruled 2 occurred in the Interactive Storytelling class at the National University of Singapore, in their Communications and New Media program. This class, taught by Alex Mitchell, provided a purely non-technical audience for our system, in contrast to the mostly technical group of students at UCSC, increasing the balance of the group with a total of 41 non-technical users. Students in this class were also assigned to create a story world, were lectured on the story generation model and system, and given documentation, the sample story, and the Little Red Riding Hood tutorial. In the following sections, we describe both quantitative story data metrics as well as a qualitative analysis of our experiences teaching people to create stories using Wide Ruled.

Results

The in-classroom evaluations of Wide Ruled 2 were successful, and showed that our tool does indeed allow non-technical authors to create generative interactive stories. We analyzed the story files for all students and compared it to the sample story given to each student as part of their training. The average story world was populated with more story objects and the story structure was higher, with higher averages for goal-fragment tree depth and max tree depth. The textual output actions typically contained more variables and references to variables in preconditions. Compared to technical authors, non-technical authors typically authored stories with lower structural complexity and story world size, but on average they still had more complex and larger stories than the sample, showing their ability to create content beyond the confines of the sample story world.

Our feedback from this evaluation, however, did point out several issues with the system. Novice users had problems learning and using complex precondition constraints, understanding recursive calls to the story hierarchy, and saving and managing variables throughout a single plot fragment. This was not surprising, given that these concepts are typically only encountered in programming classes. The asynchronous interactive actions of Wide Ruled 2 were not used very much, and feedback from students indicated that the fact that these interaction actions were always available to a reader made them less useful, because most types of interactions in a story are contextually relevant to only a limited part of the story, and wouldn't make sense anywhere else. Another critical weakness in Wide Ruled is its limited ability to model large and complex story structures, as a result of its limited interface. The hierarchical list of goals and fragments became unwieldy with large story trees and many interactive actions, and complex precondition constraints and action lists were hard to handle and understand at a glance. The detailed results of our study, along with specific notable story failures and resulting authoring techniques, are fully described in our previous paper [SKO09].

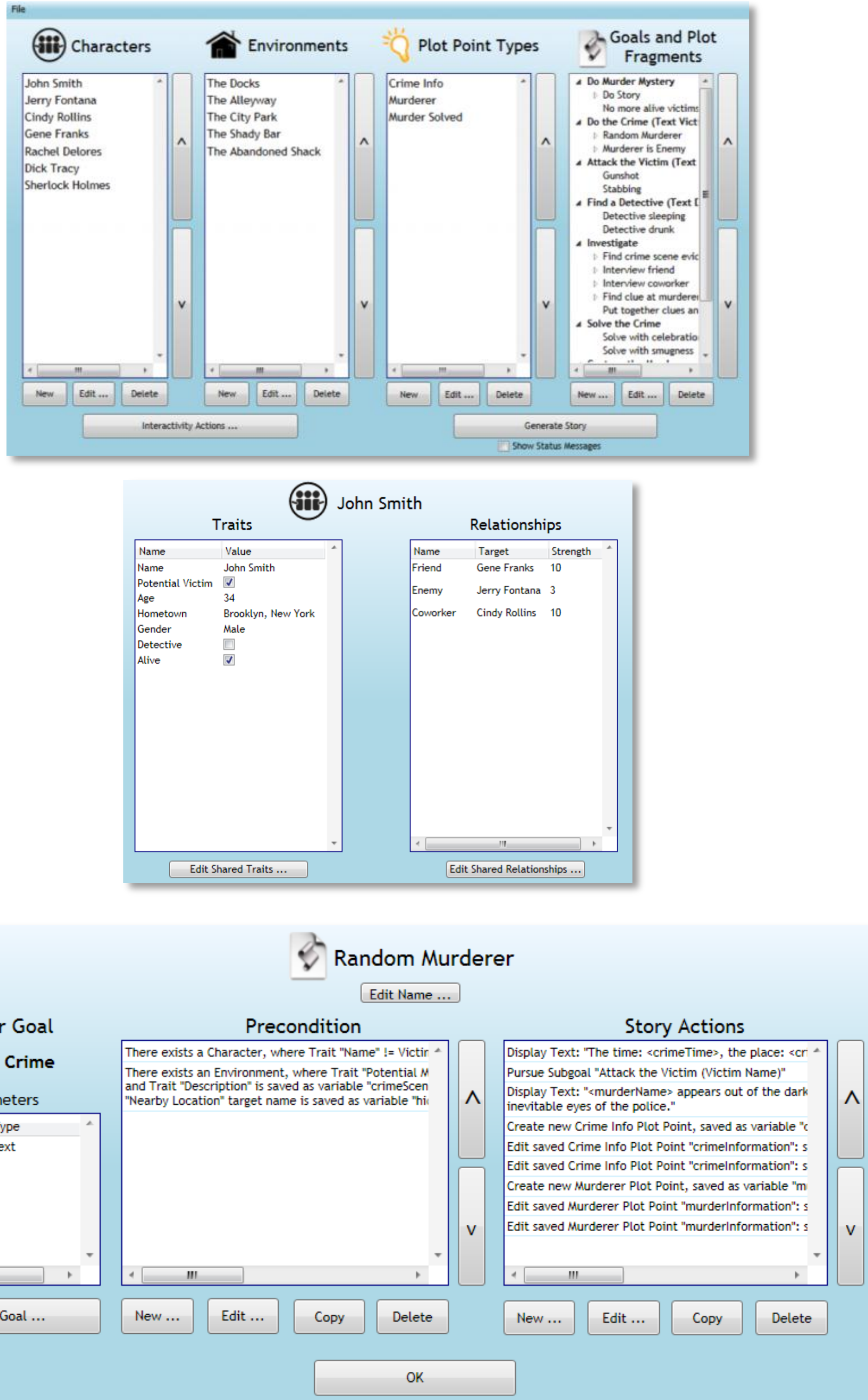

Figure 3. The Wide Ruled 2 interface, with main window (above), character editor (middle), and plot fragment editing window (below). In the main window, the characters, environments, plot point types, and story hierarchy are displayed to the author. The plot fragment editor displays the incoming author goal parameters (left), precondition constraints and bindings (center), and resulting story actions that are executed if this plot fragment is selected (right).

Story Canvas

The second stage of our research is the construction of Story Canvas, a purely graphical, storyboard-based authoring interface built atop the same ABL-driven, UNIVERSE-inspired story generation model as seen in Wide Ruled 2. It incorporates many of the lessons learned from those two systems, and makes the jump to a richer visual story representation, with an authoring interface that closely resembles the reading and interaction interface seen by a reader.

The Move to Storyboards

Wide Ruled was effectively a structured code editor for the language of UNIVERSE-based story plans, and guides a user through the step-by-step creation of each and every piece of every precondition and story action that occurs in plan. Previous research has discussed the practical problems with structured code editors, specifically that they introduce the tedium of entering code without the efficiency and flexibility of typing it by hand [KO06]. Instead of developing a complete domain-specific writer-friendly textual storytelling language, like *Inform 7* [NEL06], to address these problems and potentially introduce completely new hurdles in teaching syntax and debugging techniques, we chose to move the user further away from the underlying code and step into the visual storytelling and authoring domain. We justify this move to the visual domain by pointing to the success of domain-specific visual languages in software engineering, as described in the Related Work section. In our move to the visual domain, we chose comics and storyboards as our visualization technique because of its rich history as a successful and heavily analyzed spatio-temporal medium of visual storytelling.

Authoring in Story Canvas

Constructing a Story Canvas story world requires the author to create a set of objects upon which the story structure acts (characters, environments, plot point types), as well as the story structure itself (author goals and plot fragments). This authoring of story objects is similar to that of Wide Ruled 1 and 2, with the addition of selectable character and environment visual representations [SKO07]. Here, we will focus on the two most complex aspects of the interface: authoring the story hierarchy, and creating plot fragments. For more detail on the complete Story Canvas interface and the details of its implementation in the ABL planning language, see our previous work [SKO10].

Graphical Story Hierarchy

The interface for viewing and modifying the goal-fragment story space is shown in figure 4, which depicts a sample murder mystery story in the Story Canvas interface. Goals are represented by dotted outlines around groups of labeled plot fragments. This hierarchical layout is automatically generated and adjusted as new author goals and fragments are created and old ones are deleted by the author. In order to deal with potentially very large story trees, the interface can be panned and zoomed across the hierarchy. Here the author can create, delete, and edit goals, arrange fragments, and create empty fragments and delete existing ones. The editing of individual plot fragments is more complex and described in the next section. It is important to note that this story hierarchy is not a strict directed acyclic graph. Plot fragments can subgoal author goals recursively, represented by the upward-pointed dotted arrow connecting two author goals in the tree. In addition, a single author goal can be subgoaled by more than one plot fragment throughout the story, and therefore appear in multiple places in the hierarchy.

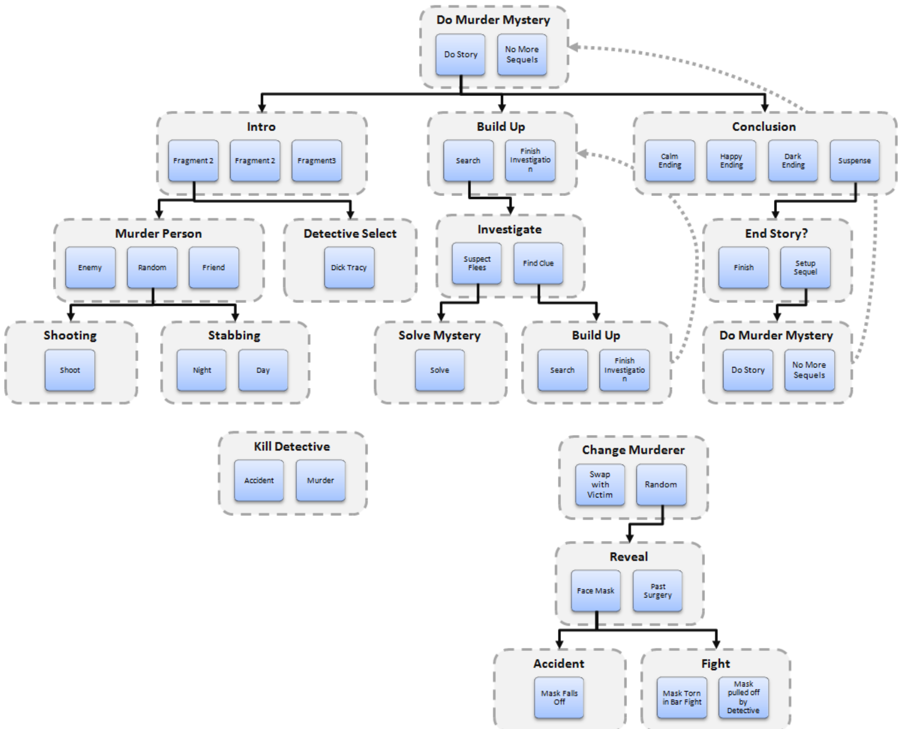

Figure 4. The Story Canvas story structure interface. Groups of plot fragments are contained within author goals, and each plot fragment can be connected to one or more author goals, represented by edges in the hierarchy. The dotted arrow depicts a recursive subgoal link.

Wide Ruled used a simple hierarchical list to depict its story structure, which proved to be unwieldy for large story spaces. The move to a zoomable and pannable visual hierarchy interface allows authors to deal with much larger story spaces. A story world in our system can be disconnected into multiple hierarchies. A single author goal is designated as the start goal (in figure 4, the top-most goal is the start goal) before story generation begins, but independent goal-fragment trees may exist in the story world at the discretion of the author. This interface makes it easy to spot distinct hierarchies (see the two small hierarchies at the bottom of figure 4). Multiple hierarchies can be a remnant of an incompletely-authored story or a failure to include an appropriate author goal in a plot fragment (a bug), or they may be intended for activation during generation by an interactive action specified by the author.

Graphical Plot Fragments

Story Canvas presents a plot fragment as a set of story panes for both the precondition and the story actions, and hides some of the complex computational constructs that caused problems for non-technical authors in our previous evaluations of Wide Ruled. Figure 5 shows a sample plot fragment being edited in Story Canvas. Precondition constraints are contained in the left-most pane, and the rest of the panes contain story actions. Below each story pane is a smaller pane, which contains abstract, non-visual constraints and story actions, that affect elements that will not be visible to a reader of the resulting story (plot point constraints and modifications, and calculations of new values are located in these panes). If an action pane only has non-visible actions, then it will not be displayed during story generation. As a plot fragment is created, the author can zoom and pan in this interface, resizing elements at will to fit into each story pane. While there is always a single precondition pane, there can be any number of story action panes in a single plot fragment.

Graphical Precondition Pane

As seen in figure 5, these constraints are represented as a graph of character and environment icons. Because precondition constraints are meant to capture characters, environments, or plot points, which are dynamically selected during story generation, their appearance is not known during the time of authoring. These "blank" icons therefore represent unknown, potential story objects, and correspond to objects placed into the story action panes on the right. Each of these captured characters, environments, or plot points are automatically given a name, which is used when referencing traits or relationships within story action panes, described later. Intra-object constraints on individual objects, such as "`age > 19`" or "`name != John`" are contained within the small box hovering next to each icon, which can be expanded into a simple constraint editor upon clicking. Inter-object constraints, which relate two dynamically bound objects, are represented by an arrow between the icons, showing that those two objects have an inter-dependency. In Wide Ruled, this inter-dependency required two separate constraint statements, in which a variable was captured and bound with a specific name, and then referenced in another statement following the first. This creation and referencing of variables has been collapsed into an edge between two icons. The box on each edge contains the various dependent constraints between two unbound story objects, displayed simply as a listed of directed comparisons between unbound object trait or relationship values. The process of naming and referencing variables within a plot fragment was a major problem for non-technical users authoring stories in Wide Ruled [SKO09]. The graph-based constraint authoring approach avoids the tedious and confusing management of variable name bindings and references. We have eliminated explicit name management by visually linking story objects to the bindings in the constraint graph.

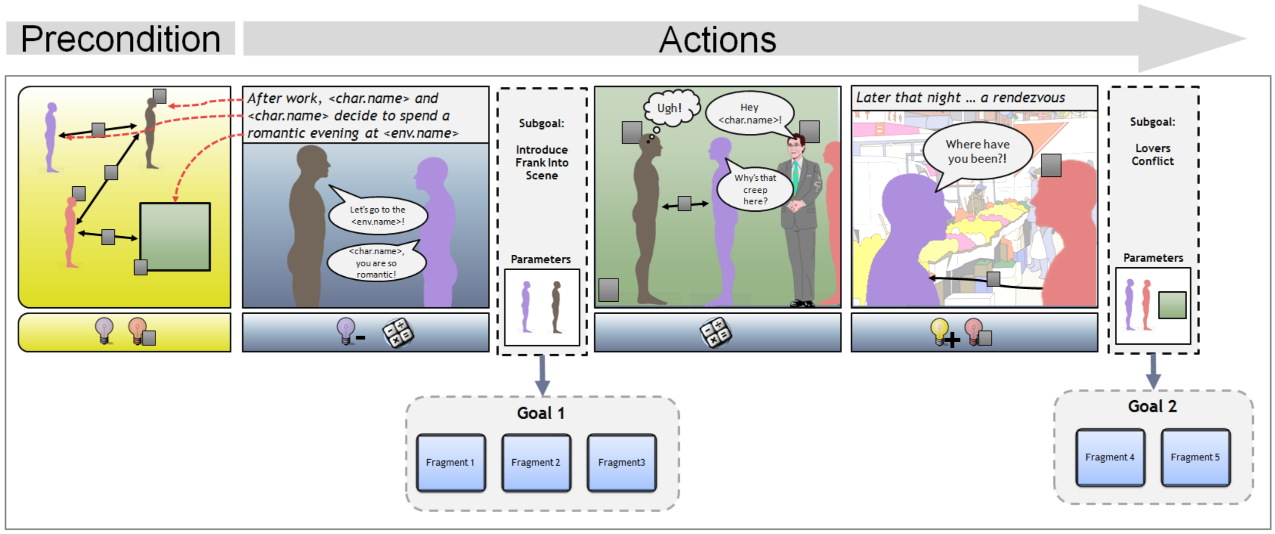

Figure 5. The Story Canvas plot fragment editor. Silhouetted objects represent dynamically-bound story objects, and images represent static elements. Dotted red arrows depict an information dependency shown when a visual element is selected.

Graphical Story Action Panes

Story action panes are depicted to the right of the precondition pane in figure 5. Actions can subgoal author goals, modify objects captured in the precondition, create/delete plot points, calculate new values, display static or captured objects with varying poses and composition, and output parameterized text in the form of narration blocks, speech bubbles, or thought bubbles. Modification of the traits of characters/environments is shown as a small box hovering next to the icon of a captured object or static image of pre-selected object. Editing of relationships between characters is depicted as a graphical link between avatars, similar to the inter-object constraints in the precondition pane. Plot points are modified (indicated by small box), created (indicated by a plus sign), and deleted (indicated a minus sign) in the bottom non-visible pane, along with the creation of calculated values. Like in the precondition pane, dynamically bound characters are displayed as colored silhouetted icons, and statically chosen objects are rendered with their associated image directly in the authoring interface. Parameterized text can be displayed as a narration block (as seen in the first and last story panes in figure 5), a thought bubble, or a speech bubble. The locations, sizes, and orientations of captured and static story objects, narration, speech and thought bubbles, as well as text size, are customizable by the author, to allow for varied and interesting compositions.

In Wide Ruled, story actions were strictly ordered lists of actions. In the domain of storyboards, actions exist on a 2D plane, and may not clearly reveal the underlying linear ordering of the actions in the ABL story plan, which can affect the resulting displayed information. In order to deal with this ordering ambiguity, Story Canvas enforces the rule that parameterized text, calculations, and modifications to objects may only reference information contained within previous panes. Subgoaling in a plot fragment is displayed as a narrow story pane with a dotted outline containing an author goal name and the story objects to be passed as parameters to that author goal. The author goal to be subgoaled, along with its contained plot fragments are shown on screen below the current plot fragment for quick access to the editing views for those elements.

Like in the precondition pane, we hide variable bindings and references where possible within the story action panes to simplify the authoring process. Whenever a story action item needs to use the information stored in a captured story object (or value created in a previous pane), the author double-clicks on the object or calculation to be edited and they can then select from a list of available information: previously calculated values, any trait or relationship value within previously captured or used story objects, or any attribute of a captured plot point. The author selects the required piece of data from a list of icons and attribute names that appear during editing. Each of the captured story objects is given an automatically generated name like “`character1`” which can be customized by the user. This allows the selected traits or relationships to be easily identified at a glance as “`character1.name`” or “`character2.name`”, with text coloring matched to that of the silhouette in the precondition pane to make it clear where the information comes from. In addition, this data dependency is then visually represented as a red arrow to the referenced object or calculation. These arrows, shown stemming from the narration text box in the first action pane of the fragment in figure 5, are only visible when an object/calculation is selected for quick informational purposes, and disappear when not selected to avoid visual cluttering.

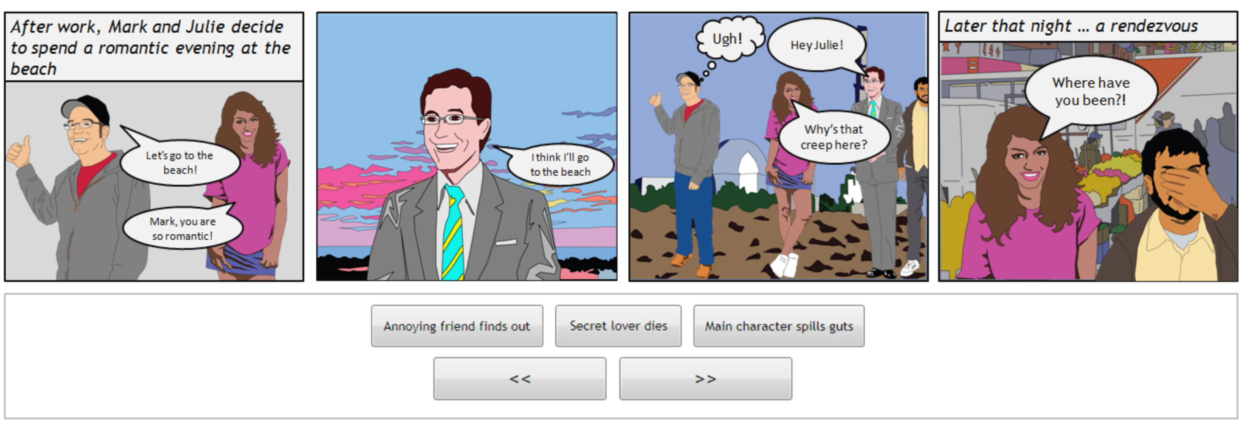

Figure 6. The Story Canvas reading and interaction interface. A generated story appears as a single comic strip that is navigated from left to right by a reader. Interactive actions appear as buttons below the comic strip. In this figure, the previously shown plot fragment has been instantiated in a generated story

Storyboard Reading and Interactivity

The reading and interaction interface for a generated story is very similar to the authoring interface, as shown in figure 6. The plot fragment shown in figure 5 is displayed to the reader, with unknown icons filled in with characters and environments, and object modification, information reference arrows, plot point modifications, and calculation actions hidden from view. This interface displays a continuous comic scrolling from left to right until the story is finished, using the arrow buttons to control the generation pace and browse previous panes.

As discussed earlier, Wide Ruled 2’s asynchronous interactive actions were maintained as global list ready to be activated at any time during story generation. User feedback indicated that creating global interactions that make sense at every point in a story is difficult, so Story Canvas allows the author to associate interactive actions with specific plot fragments [SKO09]. The process of authoring these interactive actions is the same as authoring any regular type of author goal and plot fragment in the story hierarchy. Those actions only appear during the execution of their associated fragments, as buttons that can be pressed below the story panes.

Proposed Work

Story Canvas Evaluation

Like Wide Ruled 1 and 2, we plan to evaluate the effectiveness of Story Canvas as an interactive story authoring interface for non-technical, novice authors. Like our previous studies, the system will be introduced in the UCSC Interactive Narrative class in the winter quarter of 2010, and the results of this real-world usage will provide critical feedback on the first iteration of comic-style interface metaphors and visualization methods we have implemented. In our transition to the visual story authoring domain, we have made many decisions in translating computational concepts into graphical representations. These decisions were based on our experiences with other authoring systems and our judgment as interface designers, and they may conflict with the authoring techniques, and interpretations of the novice users who will be using our system. The purpose of this evaluation will primarily be refinement of Story Canvas from an experimental evolution of an interface into a practical authoring system.

Story Canvas 2

The next version of Story Canvas will, first and foremost, address any of weaknesses in the authoring experience of the previous version. From our previous experience with new authoring software, we expect there to be necessary changes to be made to ensure that Story Canvas is an intuitive authoring environment. Next, we will attempt to increase the authorial power of the system by introducing intelligence into the interface that manages complex visualization techniques, supports automatic composition tasks, and provides real-time feedback on the complexity of the potential stories generated by the tool.

Feedback and Story Space Management

Story Canvas 2 will inform authors and assist them in managing large story plans, by executing an author's partially completed story plan in the background, rapidly and repeatedly, tracking the distribution of potential story plans across the entire potential story space. Using heuristics, the interface will then evaluate these execution runs and determine if the author's plan will potentially loop indefinitely, or likely never generate in a particular area of potential story space. These repeated runs will then generate colored heat maps, as seen in figure 7, that track and visualize plot fragment selection frequencies and help the author determine problem areas of his or her story plans that do not achieve desirable coverage. In addition, Story Canvas 2 will also attempt to let an author manage large story spaces more efficiently by collapsing entire subtrees of the story space. Two possible ways of collapsing these subtrees are shown in figure 7. This collapsing of irrelevant subtrees will be able to be triggered manually on a per-node grouping, and will also be automatic based on what part of the story hierarchy is currently being edited. There is potential for this collapsing process to intelligently group similar subtrees and allow the author to display only related regions of story space, collapsing all other nodes. We will investigate various ways of detecting similarity in the story space, such as by matching precondition constraint binding patterns and usage of similar traits or relationships between story objects.

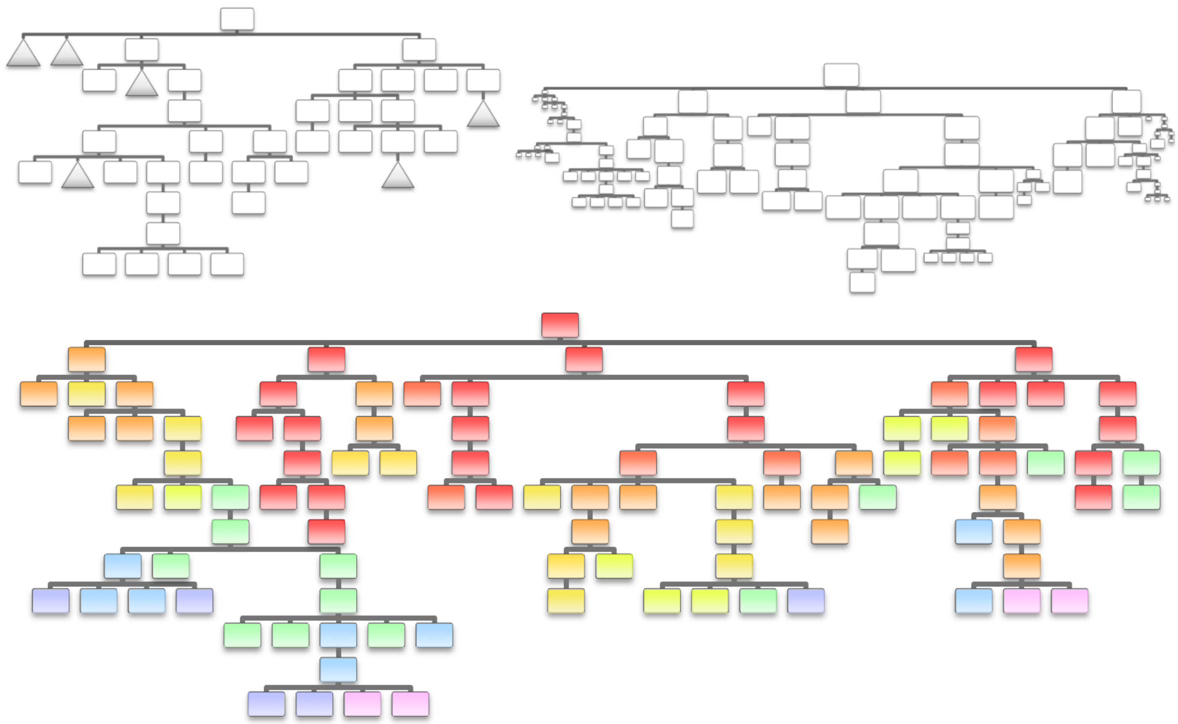

Figure 7. Methods of managing large story spaces in Story Canvas 2. Shown here are simplified goal-fragment story hierarchy representations of the same story space. Large story structure can be managed by manually or automatically collapsing subtrees into representative icons (top-left), variable node rendering sizes (top-right), and generating heat maps of our planner coverage of the story space (bottom). In this figure, red nodes are most often visited by the story generator.

Pane Modifiers

Another goal of this next iteration of Story Canvas is to continue to explore the broad space of visual techniques known to storyboard and comic authors and hone them based on user-studies and experimentation. We therefore plan to introduce the ability to modify the composition or appearance of story action panes based on information captured in the precondition pane. An example of this sort of pane modifier would be a color saturation decrease and contrast increase based on the emotional state of a character captured in a precondition. In terms of composition, an automatic zoom features that focuses on tightly framing all of the characters in a single story pane would be useful in automatically responding to the emotional distance between characters, or to convey the privacy of a conversation, simply by adjusting the scaling of the scene dynamically based on the progression of the story. Examples of these types of pane modifiers can be seen in figure 8. The specific types of stylistic modifications we implement will be inspired by the extensive analysis of comic storytelling techniques that exists [MCC93]. By implementing these computationallydriven forms of artistic techniques in Story Canvas, we hope that authors may harness these effective components of comic-based storytelling and, as a result, create more compelling interactive stories with even more variation in the reader experience.

Figure 8. Pane modifiers in Story Canvas 2 will provide computational support for stylistic changes, such as color palette modification (left), and composition (right).

EVALUATION

As a more mature, refined authoring system, Story Canvas 2 will be evaluated as a direct competitor to the established Wide Ruled 2 authoring system. We will put both tools in the hands of a population of novice authors, and instruct them to author similar stories, in order to compare the process of learning to use these systems, as well as the quality of the resulting stories made with them. Ultimately, we wish to compare the creative potential of Story Canvas - it is our hope that Story Canvas, by its very nature as a visual authoring tool with a familiar medium of authoring and storytelling, will inspire authors to create more complex and richer interactive experiences than are typically made with Wide Ruled's traditional interface and text-based output. Like our previous evaluations of Wide Ruled 2, we will use a combination of qualitative questioning and story evaluation alongside quantitative story data analysis to determine the results of this study.

A SOCIAL GAME-DRIVEN INTERACTIVE STORY AUTHORING SYSTEM

INTRODUCTION

The final stage of our work concentrates on applying the techniques we learned in the development of both Wide Ruled and Story Canvas, and combine those methods with an intelligent authoring environment that actively collaborates with the author to create interesting interactive story experiences. While Story Canvas 2 proposes introducing passive feedback to the author in the form of an evaluation of the story space, the story generation model in our final system will play an active role in helping the author construct compelling interactive narratives by reasoning over the vast space of potential social game progression, providing feedback on potentially interesting story events and ensuring reachable story states. In this section, we will discuss the social game model that will replace the UNIVERSE-style social model of previous work, explicate the necessity for and implementation of a forward-search planner on top of our social game representation, and describe two potential models of authoring and interaction based on this modified story generation model.

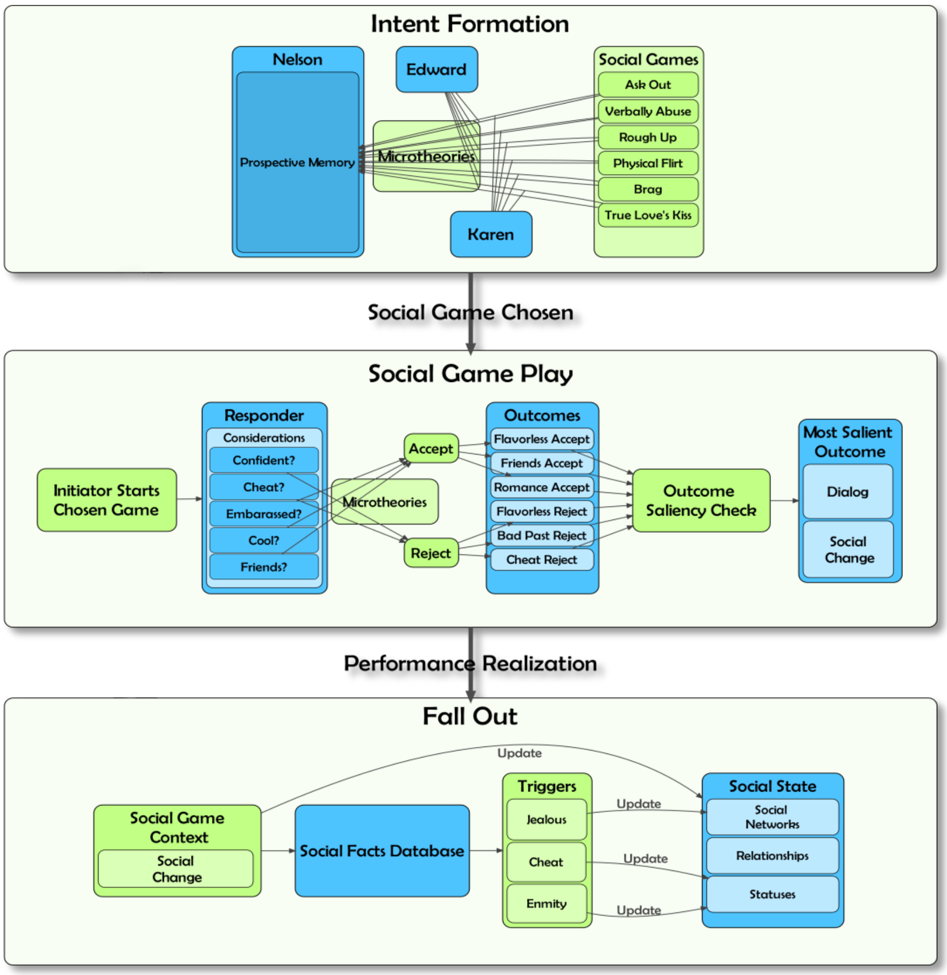

Figure 9. The *Comme il Faut* social game model [MCCO10].

COMME IL FAUT – A 'SOCIAL PHYSICS ENGINE'

We will implement this authoring system by utilizing the technology of *Comme il Faut (CiF)*, a system we discussed in the Related Work section, created as a computational model of social interaction. In contrast to the UNIVERSE model of story generation, *CiF* is a character-focused model of generation, where each character formulates goals, or volitions to perform "tasks" in the social world based on their internal qualities as individuals in the system, as well as their relationships with the other characters, and their status amongst these peers [MCC10]. The "tasks" in the world of *CiF* are the initiation of social games that have a desirable outcome, or make progress towards that outcome. Many of the knowledge constructs in *CiF* are inspired by its implementation domain: an interactive game called *The Prom*. This game, set in the dramatic teenage world of high school, was built to test the capabilities of *CiF* and demonstrate it as a practical model for creating interactive narrative. As a result of this domain choice, there is a concept of objective knowledge of “coolness” amongst peers which effectively acts as a central knowledge blackboard amongst all of the characters that has effects everywhere in the social game space. In its current form, *CiF* is a “social physics engine”, in which social games are the unit of change in a social world. It does not have a sense of high level global social progress or plot formulation, and as a result, *The Prom* has an external, simulation–focused mode of interaction in which players explore the various combinations of social games and their effects available amongst the characters in the world, with general goals but no guarantee of achievability of these goals or reward for reaching them. We aim to utilize this model in our work and apply a higher level structure to the stories it creates, thereby taking advantage of the vast space of social exploration inherent in the generation model for both the authoring and playing experience [MCC10].

Characters: Traits, Relationships, Networks and Statuses

Characters in *CiF* are composed of traits and relations to other characters. Relations are defined by binary *relationships* (*friends*, *enemies*, and *dating*), as well as a network of weighted connections, called the *social network*, between characters that numerically describes friendship, romantic affection, and respect amongst peers [MCC10]. In addition, characters have *statuses*, which are temporary binary social effects that result from social games, such as “angry at”, “popular”, and “jealous of”.

Global Social Knowledge: Social History and the Cultural Knowledge Base

As mentioned earlier, *CiF* maintains a global knowledge base of social information that is modified by the effects of social games, and that is also taken into account in some of these games. The *Social Facts Database (SFDB)* stores a history of previously executed social games and their outcomes, as well as previously discussed topics of conversation. This information is taken into account when social games are chosen, and provides for a more dynamic resulting experience in social games that take into account this social history. The potential topics of conversation which these games can utilize is stored in the *Cultural Knowledge Base (CKB)*, which contains an objective rating, shared amongst all characters of various discussion topics and their “coolness”, as well as the interest each character has in each of those topics. Like the *SFDB*, the *CKB* plays an important part in *CiF* as a shared *zeitgeist* of the story world which social games can utilize if desired.

Social Games

Social games are the core of the *CiF* experience, and can occur between any two or three sets of characters in the story world. They contain a series of preconditions that can reference all of the types of character-specific and global social state previously mentioned. These constraints apply to specific roles in each game – an initiator, a responder, and optionally an “other” character who can be brought in to participate in the conversation, sometimes taking “sides” with one character against another. If a social game is eligible for execution and selected by a character, *CiF* then evaluates (scores) the many social aspects of the initiator and his or her relationship to the responder and possibly the “other” character, combines this information with existing *SFDB* and *CKB* knowledge, and uses this score to determine the actions of the responder (the *outcome*). A single result, of possibly many potential outcomes, is then selected based on this evaluation (the *outcome saliency check*), and various pieces of information that were taken into account during the evaluation of the response is used to instantiate parameterized dialogue between characters, resulting in interaction that is based on character state, past history, and global social knowledge. The character states, networks, statuses, *SFDB,* and *CKB* are then updated as a result of executing the social game. During this process, if specific social historical patterns are matched in the *SFDB,* then triggers are initiated that summarize these patterns as statuses and relationships that can later be used in other social games. Figure 9 provides a summary of this process. More details on the system and components seen in the diagram are in previous work [MCC10].

'The Prom' as a Storytelling Domain

As we mentioned in the beginning of this section, *CiF* is implemented as the foundation of the interactive game *The Prom*, as seen in figure 10. This game contains a vast set of expertly authored social games and related knowledge that we will utilize as a planning domain in our research. While the authors in our system will create their own characters and their social histories, they will utilize the existing types of traits, relationships, statuses, social network links, and *CKB* topics. In this way, authors will be operating with a rich and deeply defined domain of interesting social interactions, and as a result, our authoring system will be primed, with some additional implementation mentioned in the next section, to suggest a large of array potential social possibilities with a high amount of story variation and consistent quality.

Figure 10 – *The Prom* is a game built to demonstrate the capabilities of *Comme il Faut* social interaction model. Our work will utilize the social games and story environment of *The Prom* as a basis for our mixed-initiative authoring system. In this screenshot, two characters are selected and the available social games and relevant relationships, traits, and social networks are displayed to the player.

Planning over Social Games

In order for our authoring system to provide feedback on future events in the potentially vast combinatoric story space of possible social games, we need to be able to reason over the effects of social games on the story world. For this reason, we will be implementing a planner on top of *CiF* that treats social games as plan operators over social knowledge. Due complex side-effect heavy nature of social games and their possible resulting status trigger rules, as well as the numerical scoring model of outcome saliency calculations, we will implement a Heuristic Search Planner, which is already well-suited to the forward-simulation style of the *CiF* social state evolution model [BON97]. By having a planner customized for this search space, we will be able to perform goal-driven searches over the space of social games, and additionally be able to verify a potential goal state is reachable.

MIXED-INITIATIVE AUTHOR-TIME PLANNING: AUTOMATIC BRANCH GENERATION

Given the combination of a planner and the *CiF* model of social games, we propose the implementation of an interface that allows authors to create, in cooperation with a planner, vast Story Canvas-like story worlds without needing to explicitly author every region of potential story space. By focusing on assisting the author in continuously delinearizing pieces of their story into a space of potential events, we aim to harness the domain of social games in order to create richer, more dynamic interactive stories with a manageable authoring load.

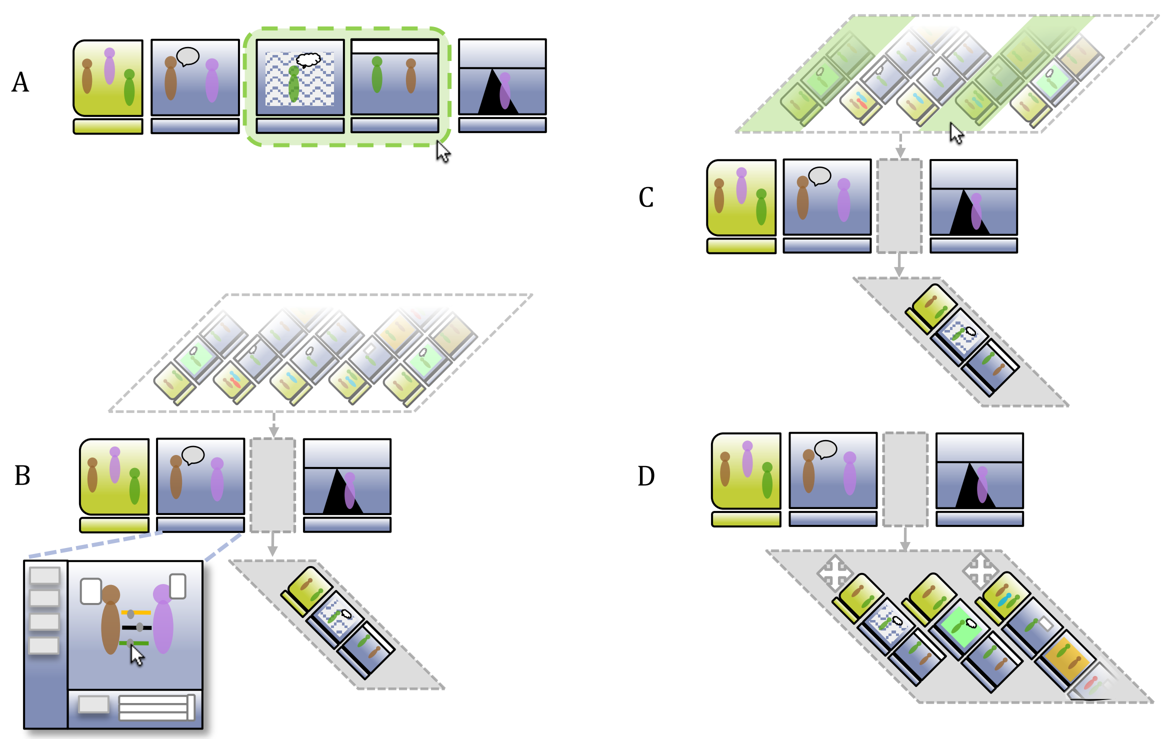

Figure 11. A mixed-initiative plot fragment editing interface. Here, an author selects a group of story panes to delinearize (A), resulting in a generated subgoal and plot fragment below, with a generated set of suggested social game based plot fragments appearing above (B). The author can explore the space of suggestions in real-time by tweaking the social state of previous story pane in the interface in the lower left of (B), and watching how the suggestion list changes as a result. Next, the user can select the desired suggested plot fragments (C) and insert them into the previously created subgoal (D). If desired, the planner could use the story pane after the selected group as a target goal during the generation process. In (D), the arrow icons next to plot fragments indicated author-selected explicit player-driven branch points.

Authoring Experience

An author using our proposed system will be presented with many of the same basic components of a story that exist in our Story Canvas story model: characters with internal state and relationships to others, author goals, and plot fragments with associated preconditions and story actions, including subgoals. The authoring of characters and their related social state in this system will, like Story Canvas, be a matter of choosing artwork, selecting traits, and assigning relationships. However, these attributes will follow the character model of *CiF* instead of the simpler model of Story Canvas 1 and 2. As a result, an author will construct traits, relationships, statuses, social network weights, components of the *SFDB* and connections in the *CKB.* In order to aid novice users in creating rich characters in the large social knowledge base, authors will be able to be instantiate new characters from templated stereotypical roles that are typical of a high school environment, to spur creativity and provide a basis for the types of characters that will fit well into the authored social games. Authors will then be free to modify those template-based characters to their liking. Construction of an initial *SFDB* will also be supported, with a set of templated history elements, such as past arguments, previous relationships or messy break-ups that you can instantiate into the database for a single character or between any pair of characters. This allows a novice user to quickly fill the social history with information that could be used by social games. The *CKB* will contain the established domain of discussion topics, and the author will use a simple graph editing interface, not unlike the precondition editor in Story Canvas, to relate characters to various "cool" and "uncool" topics in story world.

Characters authored in this new system will be considerably more complex that those of Story Canvas, with many types of connections to other players, topics of discussion, and a history of previous social events. In order to avoid overwhelming or confusing novice authors, our authoring interface will provide support for suggesting new social knowledge about existing characters, or completely new templated character types that would allow more potential for story variation. This capability will be enabled by the social game planner, which will detect over-constrained regions of the story hierarchy and respond with these types of character suggestions. We will discuss this process in more detail in the next section.

Authoring the Goal-Fragment Hierarchy

Our proposed mixed-initiative authoring system will allow authors to create story with a structure that uses the same components of the UNIVERSE-based model in our previous work: a hierarchical tree of author goals and plot fragments specific potential variations on plot progression by representing a tree with many concrete story paths that could appear during generation. While the structure of the story world in this system will be similar, the process of constructing this world will be heavily supported by the *CiF*-driven planner we have proposed.

Figure 11 shows an illustration of a proposed mixed-initiative interface for plot fragment editing. It is similar to the Story Canvas plot fragment editor in figure 5, except with a new subgoal generation and selection feature. The author would begin with a manually-authored Story Canvas style plot fragment, specifying precondition constraints (utilizing *CiF*-specific character attributes, relationships, and external social knowledge), and a set of story action panes that can perform all the types of operations that one finds in a Story Canvas pane. In addition to being able to insert modification operations that change character traits and relationship attributes, these story panes would also be able to access and change the contents of the *SFDB* and *CKB.* In general, plot fragments would be allowed to create arbitrary dialogue and modify social state of the world arbitrarily, and not be required to be consistent with the social goal setting and execution model built into *CiF*. The interface for authoring these plot fragments would mirror closely the interface for creating Story Canvas plot fragments, building off of our existing experience with storyboard-based interfaces. After an initial plot fragment had been created, the other would now utilize the mixed-initiative planning capabilities of our system. He or she would select one or more story action panes (outlined in green in section A of figure 11), and ask our system for narrative alternatives to this piece of linear plot. This user action would result in an automatically generated subgoal pane and a link to a newly created author goal (section B of figure 11), inserting the selected group of story panes into a new plot fragment within this new author goal. This new plot fragment would be passed the necessary character information bindings to generate its existing story panes. Next, the interface would then call upon our *CiF* planner to suggest possible social game-based plot fragments that could happen at this point in the story.

The process of suggesting new potential social game-based plot fragments to the user requires that our planner examine the existing authored story hierarchy to determine what social game preconditions would be potentially satisfied when executing the story hierarchy, given all of the existing characters in our story world. By exploring these various initial character selections and searching this hierarchical goal-fragment space, the planner will be able to determine whether any available social games exist, and then present these options to the user. If there exists a large number of potential social games to execute, the interface will prioritize the ones to display to the author by a heuristic of "interestingness", which will be defined the amount of intra and inter character social state change that would occur for each social game, if it were chosen. An author can select any number of these potential social games, which will then be converted into plot fragments within the current subgoal action. The preconditions of these generated subgoals will match the precondition of the social games that they originate from, and the dialogue and social state change will be mapped into a series of story panes. Once these social game selections exist as plot fragments in the story world, their preconditions and story actions could be arbitrarily modified or refined, without conforming to the *CiF* social game rule model, allowing an author to customize their story to their own liking. Given this model of interaction, the author and our system would collaboratively generate the story space, where the author creates an initial story state, the system suggests a branch in the story space, the author selects and customizes those branches, and the system in turn suggests further branches for each of those new plot fragments. By alternating the authorial burden between author and system, a potentially large story space can be explored and more easily created, resulting in more dynamic interactive stories.

Exploring the Social Game Search Space

As mentioned in the previous section, the social game planner in our proposed system would suggest a set of social games that could be executed based on the expected state of the story. We propose that this generation process is a two-way interaction: the author should be able to respond to the suggestions given by the planner by giving it new input and seeing how the resulting social game suggestions are altered, in order to understand the effects of their story actions. In order to support this interaction, our system will present an interface (illustrated in portion B of figure 11) where an author can select any story pane before the subgoal of interest, and change the story action events (addition/removal of traits, relationships, and statuses, social network value changes, etc) for each character in that pane. As the story actions are altered, the author will see the result of these changes in the changing social game suggestions that appear in the interface. We hope that this sort of interaction with the underlying planner will help expose the workings of our system and allow an author to gain an understanding of how his or her choices directly affect the story space.

Character Suggestions

In the case that our story hierarchy has been over-specified by the author and the story planner cannot find any combination of characters that allows for any of the social games to be triggered, the author will be able to request planner suggestions for modification (creation or removal of traits, relationships, statuses, or *SFDB/CKB* entries) to existing characters, or completely new characters with various sets of required traits and relationships. The new character suggestion process will be an easier solution: social game roles and their associated preconditions can be used to suggest the characteristics of a potential new character to create. In order to possibly utilize existing characters, the system would first look at the space of existing characters created by the user, and attempt to alter character attributes in order to open up the social game space. This modification process would find the social games with preconditions that are closest to being fully satisfied, match the set of satisfied constraints with the characters that already satisfy them, and use the unsatisfied social game constraints to generate new social state on these nearly-satisfying characters.

Goal-driven Social Game Suggestion

When the author prompts our interface to generate potential plot fragments for a selection of story panes, he or she is asking: "How else can this piece of my story happen?" In the case that the author selects a portion of their plot fragment that has additional story panes after it, it is possible that the author may want various potential stories that all end up in the same social state as the story panes that occur after the selection. This would form an implicit goal for our planner, prompting our system to search for an ordering of one or more social games that would arrive at the desired goal state described in the proceeding story pane. In contrast to the previously described method of single-game generation, this mode of authoring would suggest multiple orderings of many social games to the author. When utilizing this goal-driven generation model, the author will be able to explore and tweak the search space like he or she could with the single game generation process. In order to allow this, the author would be able to modify the social state of the target story pane goal dynamically and watch how the various social game ordering suggestions changes, allowing the author to visualize how various social state target goals change the types of stories that can be generated. The interface for this interaction will be the same as described before (similar to section B of figure 11), except that the story pane that is being modified dynamically will be the target goal state that the planner is utilizing in its search process. We expect that this goal-driven generation might be the most useful for the novice author, because it allows a completely linear story fragment to be spread into a space of potential stories while enforcing its overall social state progress.

Specifying Interaction: Explicit Subgoal Branches
In this system, we will implement interaction as author-specifiable explicit story branches, represented by a selected groups of plot fragments within subgoals that will prompt the player for a selection if more than one is available to execute. During play time, the system will generate the story similarly to that of the Story Canvas model, except when a subgoal is reached: if more than one explicit branch is available, a menu appears to the player with author-described plot fragment choices such as "John asks Mary out" or "Frank insults John".

Notice that this method of interaction (planner intervention) is similar to our original Wide Ruled 1 model of interaction, which we abandoned because author goals were shown to represent a level of abstraction that a reader would not typically understand or relate to in a story made with the tool. In contrast to the heavily author goal-driven story generation model of Wide Ruled and Story Canvas, this authoring system is based on the heavily character-driven *CiF* model, which explicitly models individual character interaction. While the system we have described does not force the author to create heavily character-driven, social game-based narratives, our mixed-initiative interaction is based around providing a mixed-initiative support framework that suggests plot fragments based on this character-driven domain, and therefore it will be implicitly focused on those types of stories. As a result, the plot fragments that are created in this system will not be abstract collections of story state progression, but concrete social interactions with a basis in the real world, that can be easily described by the author and presented to any type of player.

Player Experience
As described above, the player experience in this proposed system will be based on explicit branch points that replace the automatically-selected plot fragment selection logic. During generation, the player will be presented with a stream of story panes like that of the Story Canvas reading interface, with characters instantiated and dialogue filled in with the appropriate information. Occasionally during this generation process, the player would be presented with a branch in the story space which will alter the rest of the story, and will be required to select amongst various social events in the story. While the end result of this complex mixed-initiative authoring system is what appears to be a simple branching story, the underlying story model will be represented as a vast hierarchy of potential stories, resulting in widely varying interactive experience without the combinatorial explosion of authoring every possible story branch that could be chosen by the player.

Play-time Planning: Social Gameplay as a Player-Driven Subgoal
Figure 12 displays another potential story model for generating interactive stories using a plan-driven social game space. Like our previously suggested system, this work would adapt the UNIVERSE style story generation model of Story Canvas, including precondition and story action panes. Subgoaling, however, would be replaced with the concept of *social gameplay choice points*. Like Story Canvas, a story like the one depicted in the figure is generated left-to-right within any plot fragment, where the precondition selects and binds characters from the story space. Similarly, Story Canvas style preconditions in this system will be able to use any of the social knowledge available to social games, including the contents of the *SFDB* and *CKB*, and story actions could create parameterized textual dialogue and manually modify the social state of characters. These new types of plot fragments are meant to provide concrete pieces of control over the story world, as well as

scripted dialogue that progresses a story, in between sequences of social game-based story generation. During execution, the generator would execute each story pane in order, in a depth-first manner, as was done in the UNIVERSE model. The execution process, however, would differ when executing a social gameplay choice point, but continues execution when this new type of "subgoal" is completed.

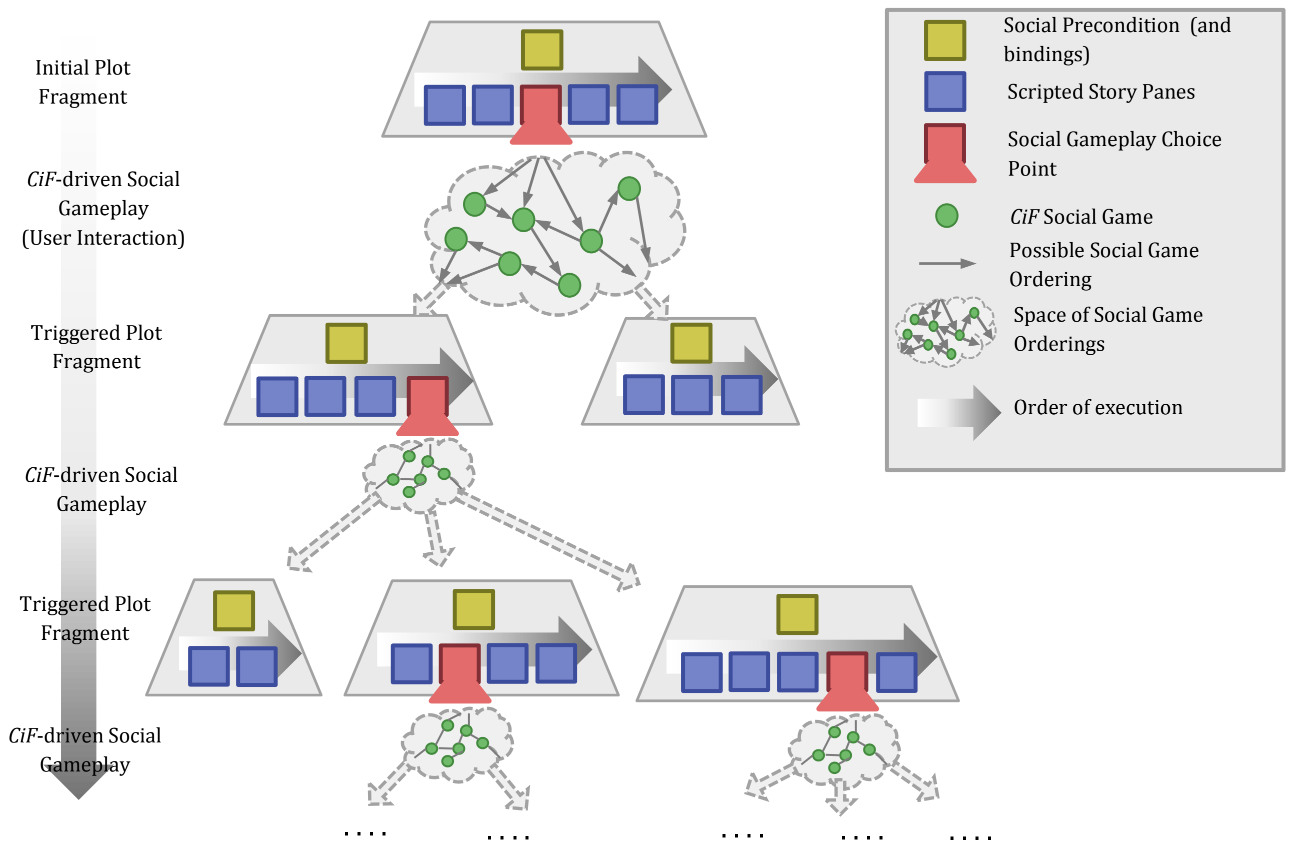

Figure 12. A model of social game planning at play time over a UNIVERSE style model. Each piece of structured story progression is similar to that of plot fragments in Story Canvas. The preconditions, in this case, are desired relationships, social network states, statuses, and social history events. The story panes are parameterized scripted story events that use the characters captured in the precondition and modify the social state of the story world. The social gameplay choice points are places where the reader is tasked with exploring the social game space established by previous scripted story panes. When a desired social state is reached, story progression can be triggered and a scripted plot fragment is executed. If desired, a player can choose not to interact, and let the story generator automatically pick a path through the space of social games.

As mentioned above, the new additions to this Story Canvas-like interface are the red-colored social game choice points. Here, the story generator enters into a character-centric generation mode, where social games must be strung together by a player to arrive at goal states (preconditions) which trigger different scripted plot fragments. In this social gameplay mode, the player can explore the space of games, initiating available social games between characters until a trigger is met (these triggers can include a maximum number of social games executed, to avoid uninteresting looping on the part of the player and to force the story into a desired state within a plot fragment). Alternatively, the player can choose to let the story generator choose the next social game to execute (or all of them), until a goal state is reached. In essence, this story generation model is a hybrid of a hierarchical goal-based system and a character-driven simulation, in which each takes turns throughout the interactive story.

In order to limit the potentially vast planning space and keep a player focused on a relevant set of characters in the world, not all possible combinations of characters and social games will be available every time a social gameplay choice point is reached. Each plot fragment will establish a cast of characters (critical initiators and responders) that can act in the social games. In order to ensure that goals are more likely to be reached, external "other" may be selected from the story world by *CiF* for three person social games, but these characters will not be selectable as characters that can initiate social games.

Player Experience

A user playing an interactive game made with our proposed system will experience the story much like a reader experiences a Story Canvas story. A stream of generated story panes will be displayed to the user, and occasionally, when a social game choice point is reached, the player will be able to begin interacting with the system, exploring the choices available for each character. The player will be presented with various options for initiating social games between any pair of selected characters that are available for the current choice point. These options, as mentioned before, will be limited by the planner in order to ensure progress towards trigger states. The player, after choosing two characters, can then choose the social game to play the resulting story panes will be generated with instantiated dialogue. Alternatively, the player can click the "next" button and at any point and two characters will automatically be selected by the generator and complete a social game. A "fast forward" button will repeat the automatic generation step until the next trigger point.

Authoring Experience

The authoring experience for this system will be similar to that of Story Canvas, with the addition of a new method of authoring the social gameplay choice points. Like the previously discussed mixed-initiative authoring interface, the author will first create a customized set of characters with internal and external social state. In this system, the creation of a hierarchy of plot fragments (with social gameplay choice points instead of author subgoals) would require an interface very similar to that of the Story Canvas system. As mentioned earlier, precondition constraint editing would use the social state of *CiF* as the basis for constraints. Traits, relationships, statuses, and social networks are all readily visualizable and authorable in the existing Story Canvas precondition pane interface. Story actions would be based heavily on dialogue and social state change. The author would still be able to create calculated values and pass information around using the same variable referencing methods as our previous work. Because we would want to allow the plot fragment component of the story model to allow as much authorial control as desired, story actions in plot fragments could create or modify social knowledge that may not have ever happened if the *CiF* social game model of change was enforced.

During authoring, when an author decides to create a choice point, he or she would select a cast of characters (gathered from matched entities in the precondition pane or statically selected from the global character list) that would populate the social gameplay choice point as initiators and responders in the social games to follow. Once these have been selected, the author would begin to create desired trigger goals/preconditions for the next part of the story. Each time one of these precondition constraints is created, the planner would attempt to evaluate whether there exists a generated story that can reach the desired trigger. If any number of paths were found, the quantity of plans would be represented by subtle blue glow to the border of the plot fragment proportional to the accessibility of that plot fragment (which look like trapezoidal bordered shapes in figure 12). If no path was found, the author would be passively notified by a red-hued pulsating plot fragment frame. If the author is finished editing the precondition and wishes to address the conflict, he or she may double click on the precondition frame. The planner would then suggest possible precondition constraints to remove in order to make the plot fragment reachable. This suggestion could be informed by a heuristic for removing "less interesting" constraints, and keeping as many "interesting" ones, based on their potential for dramatic interaction (for example, a constraint that requires a character have a strong dislike of another character might be kept because it could result in an interesting, high-intensity conflict). In the case that the planner cannot find any possible paths even after removing all constraints, it will go to the previous parent plot fragment and attempt to suggest changes at that level, and so on until the currently selected plot fragment is reachable. In this way the system would actively inform you of the potential coverage of your story space, and attempt to intelligently repair over-constrained plot fragments. During this plot fragment trigger state validation, this interface could also evaluate the quality of the space by determining how often social state changes and how much. A large or constant amount of change over many paths in the search space will result in a judgement that the search space is high quality, and the interface will provide a visual notification of this rating next to each choice point. This type of high level feedback would be akin to the heat map of generated stories proposed in Story Canvas 2. In summary, this authoring interface would utilize the social game planner to provide an open-ended simulation-style social gameplay experience in between scripted plot fragment story progression. In contrast to our previously described mixed-initiative authoring system, the social game planner would simply provide verification of reachable trigger states, and not directly suggest any new plot fragments to create. It aims to maintain the simulation-driven gameplay of *The Prom*, with the addition of intermediate progression of plot elements, instead of the author-focused branch point style of our previously suggested system.

Evaluation

Regardless of which system model we choose to pursue, the final stage of our research will be evaluated alongside our previous Story Canvas system. As in our previous evaluations, we must determine the effectiveness of the interface as a practical authoring environment, and demonstrate measurable improvement over our previous iteration. Because both Story Canvas and this newest system share a common storytelling technique and a very similar authoring interface, it makes sense to determine whether the additional complexity of the *CiF*-based system and associated planner provide any true benefit to the novice author who wishes to implement their visions for an interactive narrative, or whether the new model proves to be too confusing. We also wish to determine whether novice users will utilize and interact with the additional feedback capabilities of our newest system, or whether it would prove to be a complicating hindrance rather than an empowering capability. In addition, we will evaluate the resulting interactive experiences created by Story Canvas and our final system in front of non-authors, and determine the relative quality and depth of the stories created by both tools. We hope that taking the existing Story Canvas interface and connecting it to a rich set of domain knowledge driven by a planner will ultimately allow for a more intuitive, approachable authoring experience that constantly strives to inspire the author and encourage the exploration of the space of interactive stories.

SCHEDULE

2011	*Winter*	Evaluate Story Canvas
		Begin Story Canvas 2 development
		  Iterate on design
		  Address Story Canvas problem areas
		  Implement active story feedback
		Plan design of CiF-based authoring system

	Spring	Begin CiF-based authoring system Continue Story Canvas 2 development Evaluate Story Canvas 2
	Summer	Continue CiF-based system implementation
	Fall	Evaluate CiF-based authoring system Begin Dissertation Begin Job Hunt (Get Married)
2012	*Winter*	Continue Dissertation Job Search
	Spring	Defend Dissertation

REFERENCES

[BAL07] Balet, O. (2007). INSCAPE: An Authoring Platform for Interactive Storytelling. In Proceedings of ICVS-VirtStory 2007. LNCS, vol. 4871, pp. 176–177.

[BER02] Bernstein, M. (2002). "Storyspace 1." In Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia. J. Blustein, Ed. HYPERTEXT '02. ACM, New York, NY, 172-181.

[BON97] Bonnet, B; GO. Lyrics, GO; & Geffner, H (1997). "A robust and fast action selection mechanism for planning." In Proceedings of AAAI, pp. 714–719.

[BUR96] Burstein, M.H., McDermott, D.V. (1996). "Issues in the development of human-computer mixed-initiative planning systems." In Cognitive Technology: In Search of a Humane Interface. Elsever Science B.V. (1996)

[CAS09] Cash, S.; & Young, R. M. (2009). "Bowyer: A Planning Tool for Bridging between Declarative and Procedural Domains." In Proceedings of the Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE '09). Stanford, CA. October 14-16, 2009.

[CHE08] Cheong, Y; Kim, Y; Min, W; Shim, E; & Kim, J. (2008). "PRISM: A Framework for Authoring Interactive Narratives". In Proceedings of ICIDS 2008.

[DON04] Donikian, S.; & Portugal, J. (2004). "Writing Interactive Fiction Scenarii with DraMachina." In Proceedings of TIDSE 2004.

[EIS85] Eisner, W. (1985). "Comics and Sequential Art." Tamarac, FL: Poorhouse Press.

[FER06] Fernaueus, Y.; Kindborg, M.; Scholz, R. (2006). "Rethinking Children's Programming with Contextual Signs." In Proceedings of the 2006 Conference on interaction Design and Children (IDC '06), pp. 121--128.

[GIN03] Gingold, C. (2003). "Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds." Master's Thesis. School of Literature, Communication and Culture, Georgia Institute of Technology, April 2003.

[GOB08] Göbel, S.; Salvatore, L; Konrad, R.; & Meh, F. (2007). "StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-linear Stories." In Proceedings of ICIDS 2008.

[GOB07] Göbel, S.; Salvatore, L; Konrad, R.; Sauer, S.; & Osswald, K. (2007). "U-CREATE: Authoring Tool for the Creation of Interactive Storytelling-based Edutainment Applications." In Proceedings of EVA 2007. pp. 53–58.

[IUR06] Iurgel, I. (2006). "Cyranus - An Authoring Tool for Interactive Edutainment Applications." In Proceedings of Edutainment 2006. LNCS, vol. 3942, pp. 577–580.

[JAH08] Jahala, A.; Rawls, C.; & Young, R. M. (2008). "Longboard: A Sketch Based Intelligent Storyboarding Tool for Creating Machinima." In Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS '08), pp. 386--391.

[KEL07] Kelleher, C.; Pausch, R.; & Kiesler, S. (2007). "Storytelling alice motivates middle school girls to learn computer programming." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07), pp. 1455--1464.

[KIN07] Kindborg, M.; & McGee, K. (2007). "Visual programming with analogical representations: Inspirations from a semiotic analysis of comics." In the Journal of Visual Languages and Computing, April 2007, pp. 99--125.

[KIN05] Kindborg, M.; & McGee, K. (2005). "Comic Strip Programs: Beyond Graphical Rewrite Rules." International Workshop on Visual Languages and Computing (VLC '05), Canada, pp. 5--7.

[KO06] Ko, A. J.; & Myers, B. A. (2006). "Barista: An implementation framework for enabling new tools, interaction techniques and views in code editors." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), pp. 387--396.

[KRI10] Kriegel, M; & Aylett, R. (2010). “Crowd-Sourced AI Authoring with ENIGMA.” In Proceedings of ICIDS 2010.

[KRI07] Kriegel, M.; Aylett, R.; Dias, J.; & Paiva, A. (2007). “An authoring tool for an emergent narrative storytelling system.” In Papers from the AAAI Fall Symposium on Intelligent Narrative Technologies, Technical Report FS-07-05, pp. 55–62

[KRI072] Kriegel, M.; & Aylett, R. (2007). “A mixed initiative authoring environment for emergent narrative planning domains.” In Proceedings of the AISB Annual Convention, pp. 453–456

[LEB85] Lebowitz, M. (1985). Story Telling as Planning and Learning. Poetics 14, pp. 483--502.

[LEB84] Lebowitz, M. (1984). Creating Characters in a Story-Telling Universe. Poetics 13, pp. 171-194.

[MAC09] MacLaurin, M. (2009). "Kodu: end-user programming and design for games." In Proceedings of the 4th international Conference on Foundations of Digital Games (FDG '09), pp. xviii-xix.

[MAS10] AlterAction. (2010). Masq. http://www.alteraction.com/. Accessed: November 2010.

[MAT03] Mateas, M.; & Stern, A. (2003). “Façade: An Experiment in Building a Fully-Realized Interactive Drama.” Game Developers Conference, Game Design track.

[MAT02] Mateas, M., and Stern, A. (2002). A Behavior Language for Story-Based Believable Agents. IEEE Intelligent Systems 17(4):39–47.

[MCC93] McCloud, S. (1993). Understanding Comics. New York, NY: Kitchen Sink Press/Harper Perennial.

[MCC10] Mccoy, J.; Treanor, M.; Samuel, B.; Tearse, B.; Mateas, M.; & Wardrip-fruin, N. (2010). "Comme il Faut 2 : A fully realized model for socially-oriented gameplay." In Proceedings of Foundations of Digital Games (FDG 2010) Intelligent Narrative Technologies III Workshop (INT3).

[MEE81] Meehan, J. (1981). "Tale Spin." In Inside Computer Understanding: Five Programs Plus Miniatures. Lawrence Erlbaum Associates.

[NEL06] Nelson, G. (2006). Natural Language, Semantics Analysis and Interactive Fiction. St. Annes's College, Oxford.

[PIZ08] Pizzi, D., Cavazza, M. (2008). "From debugging to authoring: Adapting productivity tools to narrative content description." In Proceedings of ICIDS 2008.

[PIZ082] Pizzi, D.; Cavazza, M.; Whittaker, A.; & Lugrin J-L., (2008). "Automatic Generation of Game Level Solutions as Storyboards." In Proceedings of the Fourth Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '08), pp. 96--101.

[REP93] Repenning, A. (1993). "Agentsheets: a tool for building domain-oriented visual programming environments." In Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93), pp. 142--143.

[SIM05] Si, M.; Marsella, S.C.; & Pynadath, D.V. (2005). "THESPIAN: An Architecture for Interactive Pedagogical Drama." In Proceedings of the International Conference on Artificial Intelligence in Education, Amsterdam, 2005.

[SKO10] Skorupski, J; & Mateas, M. (2010). "Novice-friendly Authoring of Plan-based Interactive Storyboards." In Proceedings of the Sixth Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '08).

[SKO09] Skorupski, J; & Mateas, M. (2009). "Interactive Story Generation for Writers: Lessons Learned from the Wide Ruled Authoring Tool." In Proceedings of the 8th Digital Art and Culture Conference (DAC 2009), Irvine, CA. December 12-15, 2009.

[SKO07] Skorupski, J.; Jayapalan, L.; Marquez, S.; & Mateas, M. (2007). "Wide Ruled: A Friendly Interface to Author-Goal Based Story Generation." In Proceedings of the Fourth International Conference on Virtual Storytelling (ICVS '07), pp. 26--37.

[SMI94] Smith, D. C.; Cypher, A.; & Spohrer, J. (1994). "KidSim: programming agents without a programming language." In Communications of the ACM 37, 7 (July 1994), pp. 54-67.

[SPI09] Spierling, U; & Szilas, N. (2009). "Authoring Issues beyond Tools." In Proceedings of ICIDS 2009.

[THO06] Thomas, J; & Young, R. M. (2006). "Author in the Loop: Using Mixed-Initiative Planning to Improve Interactive Narrative." In Proceedings of ICAPS 2006 Workshop on AI Planning for Computer Games and Synthetic Characters.

[THO062] Thomas, J. M; & Young, R. M.; (2006). “Elicitation and Application of Narrative Constraints Through Mixed-Initiative Planning.” In Working Notes of the International Conference on Automated Planning and Scheduling (ICAPS) 2006 Workshop on Preferences and Soft Constraints in Planning.

[TUR94] Turner, S. T. (1994). The Creative Process: A Computer Model of Storytelling and Creativity. Lawrence Erlbaum Associates.

[WEI05] Weiß, S.A.; Mueller, W.; Spierling, U.; & Steimle, F. (2005). “Scenejo – An Interactive Storytelling Platform.” In Proceedings of ICVS-VirtStory 2005. LNCS, vol. 3805, pp. 77–80.

[WHI97] Whitley, K. N; & Blackwell, A. F. (1997). “Visual programming: the outlook from academia and industry.” In Papers Presented At the Seventh Workshop on Empirical Studies of Programmers. pp.180-208

[YOY10] YoYo Games. (2010). GameMaker 8 (Windows Program). http://www.yoyogames.com/gamemaker. Accessed: January 2010.

[ZAG06] Zagalo, N., Göbel, S.; Torres, A.; & Malkewitz, R. (2006). “INSCAPE: Emotion Expression and Experience in an Authoring Environment.” In Proceedings of TIDSE 2006.