

# Wide Ruled: A Friendly Interface to Author-Goal Based Story Generation

James Skorupski<sup>1</sup>, Lakshmi Jayapalan<sup>2</sup>, Sheena Marquez<sup>1</sup>, Michael Mateas<sup>1</sup>

<sup>1</sup>University of California, Santa Cruz  
Computer Science Department  
1156 High Street, Santa Cruz, California, 95064  
jskorups@cs.ucsc.edu, smarquez@ucsc.edu, michaelm@cs.ucsc.edu  
<sup>2</sup>Electronic Arts

**Abstract.** We present Wide Ruled, an authoring tool for the creation of generative stories. It is based on the Universe author-goal-based model of story generation, and extends this model by providing support for reader interactivity, episodic memory elements, and an extensive graphical interface that is aimed at authors with little or no experience with computer programming and artificial intelligence planning techniques. The design of the interface is based around common interface conventions, narrative terminology, and step-by-step guidance through the creation of complex plan preconditions and actions. We also present initial user evaluation of this work-in-progress, which shows how our tool can be used by those with varying technical backgrounds to author dynamic stories with our tool. This feedback also suggests potential improvements of the interface to the underlying story representation and generation model.

**Keywords:** Story generation, author modeling, interactive narrative

## 1 Introduction

The creation of interactive story experiences, in which interaction deeply affects the story, requires both expertise in authoring (compelling plot arcs, dialog, character conflicts, etc.) and technical expertise in computational and generative story representations. Either developers must be experienced artists and technologists (which, while ideal for certain types of innovation [11], is unfortunately rare), or they must work in multi-disciplinary teams that are segmented into distinct technical and non-technical pursuits. The multi-disciplinary team approach, while appropriate for large commercial projects, is inappropriate for small-scale development, impedes rapid experimentation, and is pedagogically inappropriate for small course projects. Wide Ruled is a story modeling tool that attempts to fit in this area of cross-over between algorithms and art, by providing a non-technical interface to a generative story engine based on the decidedly technical Universe story model [6, 7].

The experience of one of the authors (Mateas) in teaching Interactive Narrative over the last few years formed the genesis of the Wide Ruled project. Many of the

students in the first few years of the course, while having some programming experience, generally had arts and humanities backgrounds rather than formal CS backgrounds. A portion of the course is devoted to surveying AI approaches to story generation and management, including classic generators such as Universe [6, 7], Tale-Spin [12] and Minstrel [17], as well as more contemporary work in interactive drama. For those students who chose to incorporate an AI approach to story generation into their course projects or masters theses, they almost always chose Universe, writing partial, simplified implementations of the Universe model and building their interactive stories on top of that [e.g. 4, 8]. For students from both technical and non-technical backgrounds, the Universe story model, with its hierarchical pursuit of multi-character story plans (plot fragments) in response to author goals, appears to offer an effective compromise between generative complexity and authorial simplicity. However, since each student project created a custom, partial (and often buggy) implementation of Universe, this strongly limited the amount of time students were actually able to spend working on the content of their experiences. This observation led to the creation of Wide Ruled, a general purpose Universe-based authoring tool accessible to non-programmers, which will ultimately support a variety of front ends for story interaction. While the version of Wide Ruled presented in this paper outputs textual stories, its execution model is generic enough to be applied to any story medium, including interactive video games.

By designing a simple visual interface to author-based story planning that makes use of non-technical language wherever possible, our authoring environment is designed to provoke a feeling of familiarity and relevance to the task of story-telling, and at the same time maintain the flexibility and power of the underlying planning engine.

## 2 Related Work

The difficulty of authoring is a common concern in interactive storytelling, for AI-based approaches in particular. Many authors have described authoring issues involved in creating autonomous characters and drama managers [e.g. 9, 15, 10]; such work, however, tends to describe authoring methodologies, rather than specific tools aimed at non-programmers. In this related work section, we limit ourselves to discussions of such tools.

A number of tools facilitate the construction of story graphs, in which the author explicitly represents possible paths through the story. Some examples include: InAuthor [1], which supports authoring geographically-aware spatial stories, AESOP [14], which provides a story-graph tool for pedagogical interactive dramas, U-Create [13], which provides graph authoring capabilities as part of a larger mixed reality authoring suite, INSCAPE, which provides graph authoring as part of a modular interactive story authoring environment [18], and SceneMaker [3], which supports more complex hierarchical story graphs with probabilistic transitions. The strength of story graphs is that they provide a readily understandable visualization of the potential story space, while their weakness is that they are a non-generative formalism, forcing the author to manually express the combinatorial possibilities in the story space, thus

effectively limiting plot-level variation. The goal of Wide Ruled is to therefore make a more generative plan-based approach accessible to authors from a wide variety of backgrounds.

Other authoring tools have provided support for more plan-like story representations. DraMachina [2] defines dramatic units in which relationships between units are implicitly specified via propositional formulas over dramatic units. The Bowman plan authoring interface provides support for mixed-initiative narrative plan authoring within a plan-space planning framework [16]; it has not yet been tested with story authors.

Jayapalan created the first version of Wide Ruled [5]. Besides the evaluation, the work reported in this paper included extending the subgoaling mechanism in the underlying generation model and substantially revising the user interface.

### 3 Design

Wide Ruled implements the Universe author-based model of planning, which formulates a dynamic story as a plan that acts within the context of a story world. The world contains *objects*, including *characters*, *environments*, and *plot points* (see below). The space of potential stories is modeled as a set of *author goals*, each of which has one or more *plot fragments* which can fulfill these goals. Plot fragments (plans), the core of the story generation process, are activated based on the satisfaction of required *preconditions*; the *actions* of a plot fragment can display dynamic story text, pursue any number of author goals, and create or modify story world objects. With this representation, the underlying story generation engine selects an initial author goal and executes valid plot fragments and their pursued subgoals to

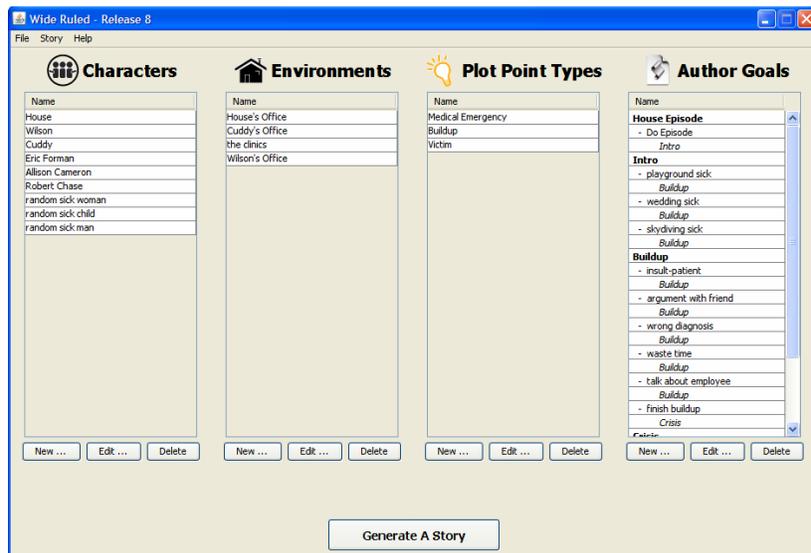


Fig. 1. The main window in Wide Ruled.

generate a single instance of story. This approach is strongly related to HTN planning where the task decompositions (plot fragments) coordinate the activities of multiple characters rather than individual characters. This is in fact the Universe model's contribution to the story generation literature, being the first system to model the pursuit of authorial plans, which coordinate multiple characters, in pursuit of authorial goals. Subgoals within plot fragments can pass parameters into the plot fragment that is selected to satisfy the subgoal, providing a mechanism for maintaining consistency in the characters and story-world objects involved in the action. By being able to represent complex hierarchies of plot fragments that satisfy author goals, Wide Ruled, like the Universe model that it is based on, can generate complex and varied stories automatically.

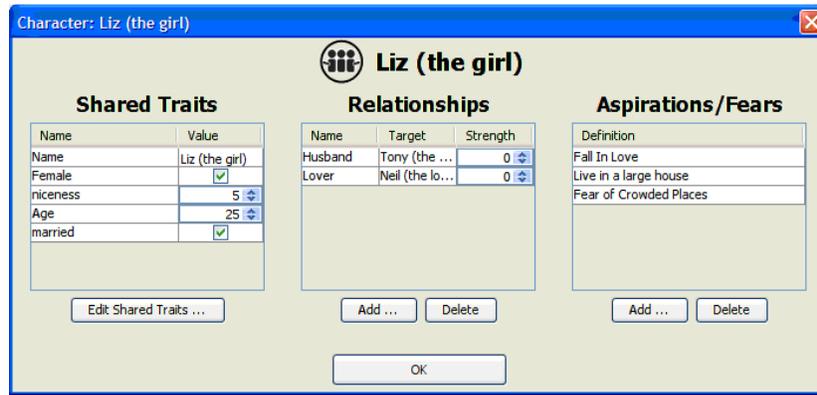
Interactivity can be incorporated in the Universe model in a number of ways, including: the third-person interaction of manipulating initial conditions of the story world, the third-person interaction of manipulating the story-world during the planning process (thwarting or aiding the currently executing author plan), and the first person interaction of making choices for one of the characters within the story world. Wide Ruled currently supports this latter interaction model, allowing a user, acting as a character within the story world, to select amongst a set of plot fragments that are related to that chosen character. The user is effectively incorporated into the planning process, making decisions among plot fragments during subgoal pursuit when those plot fragments are relevant to the player character.

Figure 1 shows the initial and main window of Wide Ruled. Here, the user is presented with three lists of named story world objects, including characters, environments, and plot point types, and a list of author goals and the plot fragments that can satisfy them. An author can perform a consistent set of operations on each list, including renaming by double clicking on each element, and creating, editing, or deleting by selecting an element and using the respective buttons below. The menu bar at the top of the window provides options for creating a new blank story world, saving an existing one to a file, or loading a previous story world from a saved file. Finally, the prominent story generation button at the bottom of the window initiates the (interactive) story generation process.

### 3.1 Objects

Objects are uniquely named entities possessing a set of user-defined attribute/value pairs. Character and event objects exist before the generation of a story, thus defining the story world. Plot points, on the other hand, are episodic memory elements that represent significant story events. They are created by plot fragment actions.

**Characters.** Characters in Wide Ruled are modified using the editor interface displayed in Figure 2. Each character has three classes of attributes – traits, relationships, and aspirations/fears. Traits are a set of named attributes that are shared amongst all characters in the story world. Each trait can be one of three types of information: text, yes/no, and numeric. Relationships are named attributes that have a target, which is a reference to another character, and a number value representing the strength of that relationship. Aspirations/fears are named attributes that, unlike shared

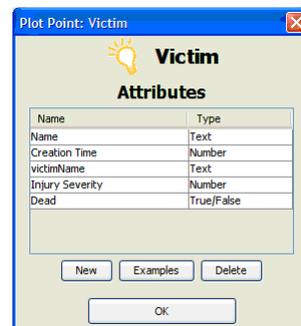


**Fig. 2.** The character editor window lists the traits, relationships, and aspirations/fears of a specific character. The add and edit buttons support the creation of new attributes

traits and relationships, don't possess any associated value, but can simply exist or not exist for each character. In order to aid in the authoring process, Wide Ruled provides many sample shared traits, relationships, and aspirations/fears that can be selected when creating new attributes.

**Environments.** Environments in Wide Ruled are analogous to characters, but conceptually refer to locations in the story world. The editing of each environment is identical to the character editing process described earlier, however only shared traits and relationships are valid attribute types, and relationships refer instead to other environments, instead of characters. Environments are intended to serve as stages for story events; attributes provide the means for selecting appropriate stages as the story progresses.

**Plot Point Types.** Plot points are episodic memory elements that are designed to represent significant events that occur during the progress of a story, and are dependent on the order of events. As such, are created and edited only during the story generation process, and the main Wide Ruled window only allows for the creation of specific types of plot points. Each named plot point type refers to a potential plot point with a user-defined set of attribute-value pairs. The values, like the values for the attributes of characters and environments, can be textual, yes/no, or numeric. In Figure 3, the simple editing interface for a single plot point type is shown.



**Fig. 3.** The plot point type editor window.

Here, the user can select only attribute names and types, and no values, since this type of object is only instantiated and manipulated during story generation.

### 3.2 Author Goals

Author goals are the primary organizational component of story generation in Wide Ruled. Like the Universe model, Wide Ruled generates stories by picking an initial author goal then executing plot fragment with a valid precondition, which may in turn pursue another author goal. As seen in Figure 1, the main window displays author goals as the top level entity of a hierarchical list. In this list, author goals are bolded elements, and each plot fragment that fulfills a specific author goal is indented underneath that goal in the list. In addition, if a plot fragment pursues another author goal, then the name of that goal is displayed in italics, indented underneath that plot fragment's name.

**Goal Parameters.** During story generation, author goals can be passed data values, either text, yes/no, or numeric, from plot fragments that choose to pursue them. These values are then in turn passed onto the plot fragment that is selected by the story generation engine for that author goal. In order to select the names and types of parameters for each author goal, the user can, from the main window in Figure 1, select any bolded author goal from the list and click the "Edit" button below. This will display a window, similar to the plot point type editor in Figure 3, which allows the user to add new parameters and their respective types.

### 3.3 Plot Fragments

Plot fragments are the unit of story world manipulation within Wide Ruled. When the story generation engine is creating a story, upon reaching an author goal, every plot fragment that fulfills that goal is analyzed to determine if all of its preconditions are satisfied. If these conditions are all satisfied, then the story generator puts that plot fragment into a list until all plot fragments are analyzed. When finished, the story generator then selects a random fragment from this list, and then performs the actions of that plot fragment.

The plot fragment editor window is activated by selecting a plot fragment name in the author goal hierarchical list in the main window of Wide Ruled. Shown in Figure 4, the plot fragment editor consists of four sections describing interactivity features, author goal settings, preconditions, and actions. The settings pane allows the user to change the parent author goal for the current plot fragment, and also displays the names and types of the author goal parameters that are passed onto this plot fragment during story generation. In addition, this pane allows the user to select whether this plot fragment can be repeated more than once throughout the course of story generation, which can occur if author goals are recursively pursued. The interactivity, precondition, and action panes are described in later sections of this paper.

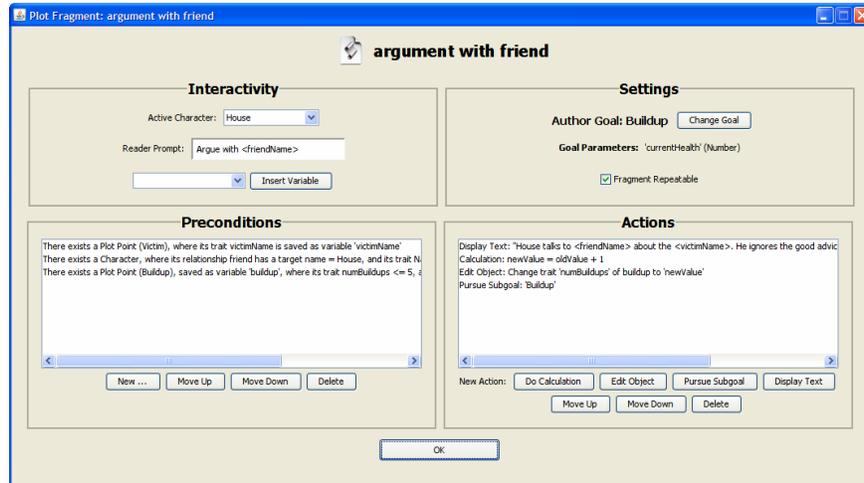
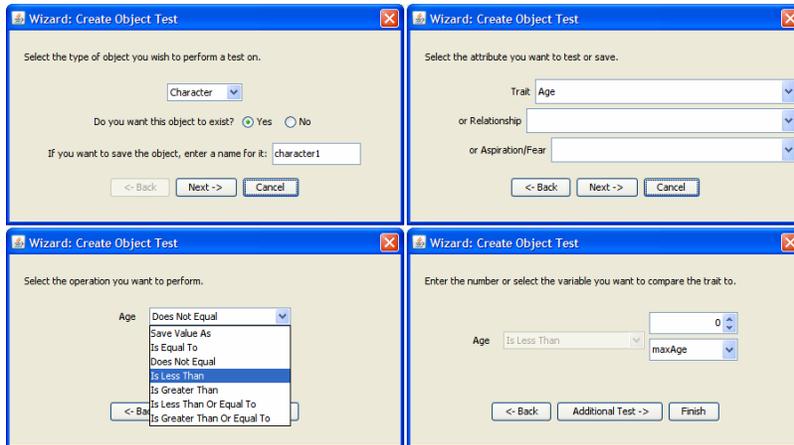


Fig. 4. The plot fragment editor window.

**Preconditions.** Preconditions are sets of constraints that must be satisfied before a plot fragment can be chosen to fulfill a goal. Within these precondition tests, during story generation, object attributes are matched against values, and any attribute of a matched object can be saved to variables within a plot fragment. These attributes may be matched against literals, incoming author goal parameters, or variables bound in previous precondition tests. A wizard is used to construct each test in a precondition, allowing the selection of the type of object to match against, attribute constraints, and options to bind attribute values to variables. This precondition creation wizard is shown in Figure 5.

Within the plot fragment editor shown in Figure 4, precondition tests are displayed as a list of simple sentences that explain, in a straightforward way, the condition that must be true for each one to be satisfied. We designed these sentences to resemble English language statements, so that non-technical users would not be intimidated by any confusing notation. The story generation engine analyzes precondition tests in the order they are listed; by selecting one and using the “Move Up” and “Move Down” buttons below this list, the user may modify this order. This order is important if one precondition test compares an object attribute against a variable that is stored in another precondition test. If analysis occurs in the incorrect order, a variable may not be bound before it is tested against, resulting in the failure of that precondition test, invalidating the entire plot fragment during story generation.

**Actions.** Actions are the core of the story generation process. They are tasks that can output story text, create new plot fragment variables, manipulate the story world, or activate other author goals to pursue. Any of the plot fragment action can make use of variables within the plot fragment. These include author goal parameters or variables



**Fig. 5.** The precondition creation wizard. The interface proceeds from left to right, and top to bottom. Any additional test or variable save for this precondition is appended by clicking the “Additional Test” button in the final window, which returns the user to the upper right window for further tasks within the current precondition test.

saved within any of the precondition tests. Like precondition tests, each action is executed in the order they are displayed in the plot fragment editor in Figure 4. Similar to the precondition list, plot fragments are displayed as a list of textual sentences that are designed to be as non-technical and understandable as possible.

*Display Text.* The text display action is the function that allows the author of a Wide Ruled story to output the actual text of the generated story. A parameterized text template language allows outputted text to depend on variable values.

*Calculation.* The calculation action allows the user to create a new plot fragment variable that is a summation, subtraction, division, or multiplication of two literal numbers, two plot fragment number variables, or one of each of these kinds of data. The ability to perform simple mathematical operations within a plot fragment allows for the flexibility of having object attributes with continuous variability.

*Edit Object.* The object editing action allows an author to create a new instance of a character, environment, or plot point, or edit any existing object. If the author wishes to edit an object, it must be bound to a variable within one of the precondition tests or created within a previous plot fragment action. The name of this bound variable is selected in the first window of the precondition creation wizard in Figure 5. The process of editing or creating objects is presented in a wizard interface similar to that of the precondition creation wizard. The wizard walks the player through assigning initial attribute values to a newly created object, or changing attributes of an existing object.

*Pursue Subgoal.* The subgoal pursuit action allows the user to select an author goal to pursue when executing the ordered actions of this plot fragment. If the selected goal requires author goal parameters, then the user must choose values to pass on from amongst the similarly-typed variables within the current plot fragment or the incoming author goal parameters.

### **3.4 Generation**

The process of story generation within Wide Ruled is a recursive depth-first descent through a hierarchy of pursued author goals. The process begins with a randomly chosen initial author goal with no parameters and at least one valid plot fragment (one with fully satisfied precondition tests), and proceeds until all the ordered actions of every plot fragment in this descent have been completed. During this traversal, if a plot fragment fails, then the story generator reverts any story world and story text changes made by the fragment's actions, and returns to the parent author goal to search for another valid plot fragment. If no valid plot fragments remain, the entire author goal fails. Failure of an entire plot fragment occurs when any author goal pursuit action within that plot fragment fails. If the initially chosen author goal fails, then another one with at least one valid precondition is pursued. While reverting on failure is possible for textually generated stories (essentially be "crossing out" text that has already been output), reversion is inappropriate for real-time visual story worlds. Future versions of Wide Ruled will explore backing-up on plan failure without reversion, using special plan "clean up" actions to maintain coherence when abandoning a story trajectory.

### **3.5 Interactivity**

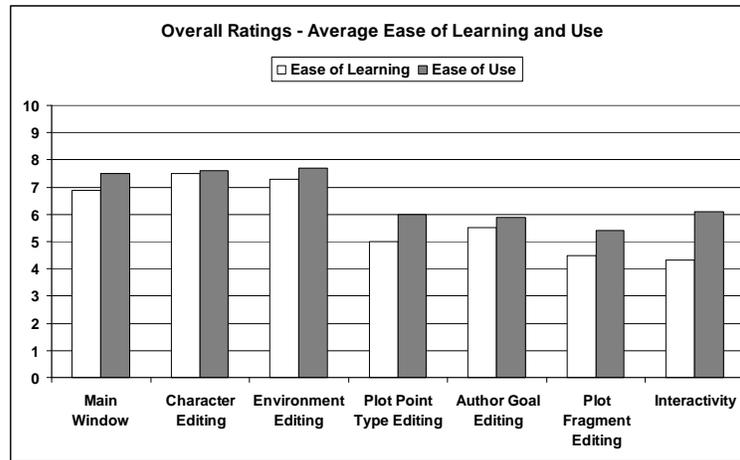
The Wide Ruled model of interactivity involves allowing the reader to override the random choice of the story generator and select which of the valid plot fragments to activate for the current author goal, if more than one exists. In order to mimic the decisions of a specific character instead of the decisions made by the story generation algorithm, each plot fragment can have a designated *active character*, chosen in the interactivity pane of the plot fragment editor in Figure 4. This active character can be one of the predefined characters created in the character editor, or it can be set to a dynamically bound character variable saved within any of the precondition tests. In order to provide a more understandable selection menu to the player, instead of simply listing static plot fragment names, the author may enter a player prompt in the plot fragment editor. The player, before generating the story, can select the character he or she wishes to play from the "Story" menu in the main window of Wide Ruled. During story generation, if the current author goal has valid plot fragments with an active character identical to the player character, then the player is presented with a set of prompts to choose from. We consider this mechanism to be a debug mechanism for exploring how abstract player choices affect story generation. In future integration of Wide Ruled with a more visual representation (e.g. game engine), rather than presenting explicit, dynamically generated menus, as is done for textual generation,

story-plan interaction would be enabled by having player choices in the concrete world *implicitly* select among potentially applicable plot fragments in the more abstract story structure.

## 4 Results

The Wide Ruled tool was presented in an assignment for an Interactive Narrative class offered by the Computer Science Department at the University of California, Santa Cruz. The assignment instructed the students to take a generative story world each had previously formulated in a story grammar, and then implement and extend the story using Wide Ruled. Ten students participated in a survey after completing their respective Wide Ruled story worlds. The students consisted of nine persons with backgrounds in computer science or engineering, and one with a background in the liberal arts. Prior to authoring, students were given a 30 minute lecture on the main features and story representation concepts within the system, four pages of electronic documentation, and two example Wide Ruled story worlds. One of them was a Wide Ruled version of the story in the popular American medical mystery drama “House”. It contained extensive manipulation of all story world objects, including multiple plot points, and varied use of preconditions, but its plot fragment structure was more linear, with each plot fragment pursuing typically only a single author goal. To demonstrate a more complex hierarchical story structure, we also provided them with an implementation of the “churn” plot fragment and its related actions, as described by Lebowitz in his work on Universe. This example demonstrated a large plot fragment with five precondition tests and seven actions that pursued other author goals, but did not make use of environments or plot points. After participants had completed authoring (they had 2 weeks), they filled out an anonymous survey in which they provided by qualitative and quantitative feedback on their experience.

The average overall ratings of the various components of Wide Ruled are depicted in Figure 6. While it appears that the character and environment editor portions of Wide Ruled received positive feedback, the responses indicate that most participants had trouble learning and regularly using the plot point type, plot fragment, and author goal editors. Not surprisingly, the editing of plot fragments, the most complex aspect of our tool, was considered to be one of the hardest tasks to learn and the hardest to use. The survey indicated that the primary difficulty was due to a lack of an ability to edit existing precondition tests or actions, or duplicate them to create similar entities with slight differences. The negative feedback to the author goal and plot fragment story structure components in general appears to demand a more intuitive explanation of the story generation process to users of the program. In terms of interactivity, the overall level of relative dissatisfaction with our model of interaction and qualitative survey feedback suggests that placing interactivity as a component of the story planning process is not a highly intuitive or conceptually simple way of integrating user input into the story generation process. In other, more positive results, two users that had low self-ratings of their background in programming and software design both reported that they used two to three precondition tests per plot fragment, the average across participants. This result suggests that the potentially complex notion of



**Fig. 6.** The overall usability ratings for each component of Wide Ruled - 1 is hard, 10 is easy.

matching multiple constraints against the story world state was presented in an intuitive manner. Unfortunately, throughout the entire set of evaluated students, many of the generated stories displayed a primarily linear plot fragment structure (low author goal count), similar to the structure of the “House” sample story world described earlier. The power of the Universe model, and thus the Wide Ruled system, is its ability to represent complex hierarchical author goal relationships. This tendency for students to write less hierarchical plot fragments did not fully exploit this flexibility, and may have been a result of bias from the earlier story grammar assignment, in which students wrote relatively abstract, fairly shallow grammars. Finally, the aspiration/fear attributes of characters also rarely used, and their task was primarily subsumed by shared traits with yes/no values.

## 5 Conclusions and Future Work

We have shown that it is possible to build a visual story authoring environment based on Universe, a classical AI model of story generation. Such authoring tools have great potential to put more powerful, generative interactive story models in the hands of story artists. Despite the mixed results of our evaluation, the results were strong enough to warrant further development of Wide Ruled, especially given our experience with non-technical students being attracted to the Universe model. Based on feedback from the evaluation, our next steps will include adding the ability to edit and duplicate existing precondition tests and actions, implementing a visual depiction of plot point manipulation by plot fragments, allowing authors to explicitly see the creation and testing of plot points, further investigation into the authorial usefulness of separating aspirations and fears from shared traits, and adding explicit support for character group affiliations. For player interaction, in addition to the active character interaction model currently implemented, we will integrate a third-person interaction

model in which the player modifies world state during story generation. In the longer term, we will explore the use of a comic strip style interface for both story generation and drag and drop authoring. Finally, with this improved version of Wide Ruled, we'll perform a broader authoring study utilizing a larger set of participants with non-technical backgrounds.

## References

1. Barrenho, F., Romao, T., Martins, T., Correia, N. InAuthoring environment: Interfaces for creating spatial stories and gaming activities. In Proceedings of the 2006 ACM SIGCHI international conference on advances in computer entertainment technology. (2006)
2. Donikian, S., Portugal, J.: Writing Interactive Fiction Scenarii with DraMachina. In Proceedings of TIDSE. (2004)
3. Gebhard, P., Kipp, M., Klesen, M., Rist, T. Authoring Scenes for Adaptive, Interactive Performances. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03), pp. 725-732. (2003)
4. Hochhalter, B.: Triad. Digital Media masters thesis, Georgia Institute of Technology. [http://steel.lcc.gatech.edu/ms\\_projects/hlogas/](http://steel.lcc.gatech.edu/ms_projects/hlogas/). (2004)
5. Jayapalan, L.: Wide Ruled: A Generative, Interactive Story Authoring Environment. Digital Media masters thesis, Georgia Institute of Technology. <http://egl.gatech.edu/?cat=4>. (2005)
6. Lebowitz, M.: Story-Telling as Planning and Learning. *Poetics* 14, pp. 483-502 (1985)
7. Lebowitz, M.: Creating Characters in Story-Telling Universe. *Poetics* 13, pp. 171-194 (1984)
8. Logas, H.: Agency: A Character-Centric Approach to Single Player Digital Space Role Playing Games. Digital Media masters thesis, Georgia Institute of Technology. [http://steel.lcc.gatech.edu/ms\\_projects/hlogas/](http://steel.lcc.gatech.edu/ms_projects/hlogas/). (2004)
9. Loyall, A. B., Neal Reilly, W. S., Bates, J., Weyhrauch, P.: System for authoring highly interactive, personality-rich interactive characters. Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 59-68 (2004)
10. Mateas, M., Stern, A.: Structuring Content in the Façade Interactive Drama Architecture. In Proceedings of Artificial Intelligence in Interactive Digital Entertainment (AIIDE), (2005)
11. Mateas, M., Stern, A.: Build It to Understand It: Ludology Meets Narratology in Game Design Space. In Proceedings of the Digital Interactive Games Research Association (DiGRA), (2005)
12. Meehan, J.: Tale Spin. In R.C. Shank and C.K. Riesbeck (Eds.) *Inside Computer Understanding: Five Programs Plus Miniatures*. Lawrence Erlbaum Associates. (1981)
13. Sauer, S., Osswald, K., Wielemans, X., Stifter, M.: U-Create: Creative Authoring Tools for Edutainment Applications. In Proceedings of TIDSE 2005, pp. 163-168. (2006)
14. Silverman, B., Johns, M., Weaver, R., Mosley, J. Authoring Edutainment Stories for Online Players (AESOP): A Generator for Pedagogically Oriented Interactive Dramas. In *Lectures Notes in Computer Science: Virtual Storytelling*. Springer (2003)
15. Sobral, D., Machado, I., Paiva, A.: Managing Authorship in Plot Conduction. In G. Goos, H. Hartmanis, and J. Leeuwen (Eds.), *Virtual Storytelling (ICVS)*, Springer (2003)
16. Thomas, J., Young, M.R.: Author in the Loop: Using Mixed-Initiative Planning to Improve Interactive Narrative. In the ICAPS 2006 Workshop on AI Planning for Computer Games and Synthetic Characters. (2006)
17. Turner, S. T.: *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum Associates. (1994)
18. Zagalo, N., Göbel, S., Torres, A., Malkewitz, R.: INSCAPE: Emotion Expression and Experience in an Authoring Environment. In Proceedings of TIDSE. (2006)