The HypeDyn Hypertext Fiction Authoring Tool

Alex Mitchell
Department of Communications and New Media
National University of Singapore
Singapore
alexm@nus.edu.sg

Kevin McGee
Department of Communications and New Media
National University of Singapore
Singapore
mckevin@nus.edu.sg

ABSTRACT

In this paper we describe the HypeDyn procedural hypertext fiction authoring tool. HypeDyn supports visual authoring of adaptive hypertext fiction in which links and nodes may be varied procedurally as the result of past reader actions. This allows for the creation of procedural stories which are much more flexible and adaptive than traditional hypertext. HypeDyn was initially created as a teaching tool, and has been used since 2009 in an interactive storytelling class in the Department of Communications and New Media at the National University of Singapore. It has also been used as a tool for research, providing a flexible platform for the exploration of issues of procedural change and rereadability in interactive stories. Our experiences with HypeDyn suggest that hypertext constitutes a valuable paradigm within which to explore issues relevant to interactive storytelling in general.

Keywords

interactive storytelling, hypertext fiction, procedural hypertext, authoring tools

Categories and Subject Descriptors

H.5.4. [**Information Interfaces and Presentation**]: Hypertext/Hypermedia

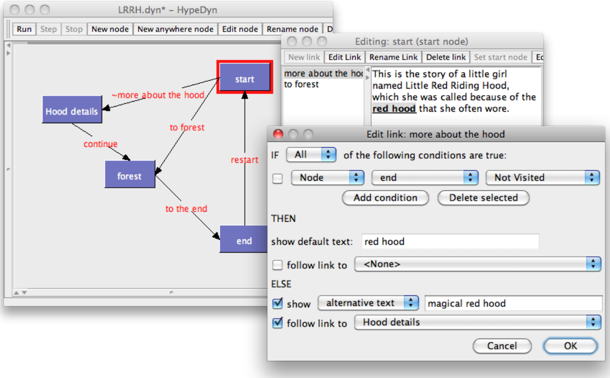
1. INTRODUCTION

The HypeDyn hypertext fiction authoring environment [6] is a tool for creating adaptive, procedural hypertext stories. It is designed to enable non-technical authors to have some hands-on exposure to the issues involved in creating procedural interactive stories. It was originally created as a teaching tool for use in an interactive storytelling class taught in the Department of Communications and New Media at the National University of Singapore. It has also been used as a development environment for our ongoing research into interactive storytelling.

In this paper, we begin by providing a brief overview of HypeDyn, and then describe the ways in which the tool has been used in our teaching and research. We end by outlining some of our planned future work.

2. OVERVIEW OF FUNCTIONALITY

HypeDyn combines the visual representation of StorySpace [3] with the conditional text and links of ConnectionMuse [5]. HypeDyn allows authors to set conditions on links, which determine whether or not a link can be followed based on the reader's past actions: which nodes have been visited, and which links have been followed (see Figure 1). This is very similar to "guard fields" in StorySpace. HypeDyn also allows authors to procedurally alter text *within* a node, based on similar conditions. This allows authors to introduce procedural change in response to reader choice without having to write code.

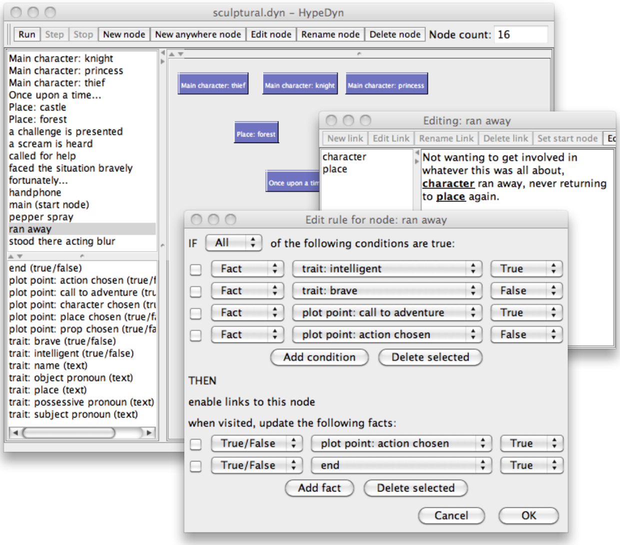


This is the story of a little girl named Little Red Riding Hood, which she was called because of the **red hood** that she often wore.

IF All of the following conditions are true:
□ Node end Not Visited
THEN
show default text: red hood
□ follow link to <None>
ELSE
☑ show alternative text magical red hood
☑ follow link to Hood details

Figure 1: Conditions on links and conditional text.

In addition, HypeDyn includes the ability to incorporate more procedural change through the use of "anywhere nodes" and "facts" (see Figure 2). Facts are variables which can be created through a visual interface, can be changed when nodes are visited or links are activated, and can be used in conditions. Anywhere nodes are text fragments which can be potentially appear anywhere in the story. These nodes can have preconditions attached to them, allowing the author to determine which "facts" must be satisfied for the text fragment to be seen by the reader, supporting the creation of sculptural hypertext [2].



sculptural.dyn – HypeDyn

Node count: 16

Main character: knight
Main character: princess
Main character: thief
Once upon a time...
Place: castle
Place: forest
a challenge is presented
a scream is heard
called for help
faced the situation bravely
fortunately...
handphone
main (start node)
pepper spray
ran away
stood there acting blur

end (true/false)
plot point: action chosen (true/f
plot point: call to adventure (tru
plot point: character chosen (tru
plot point: place chosen (true/f
plot point: prop chosen (true/fa
trait: brave (true/false)
trait: intelligent (true/false)
trait: name (text)
trait: object pronoun (text)
trait: place (text)
trait: possessive pronoun (text)
trait: subject pronoun (text)

Editing: ran away

character
place

Not wanting to get involved in whatever this was all about, **character** ran away, never returning to **place** again.

Edit rule for node: ran away

IF All of the following conditions are true:

□	Fact	trait: intelligent	True
□	Fact	trait: brave	False
□	Fact	plot point: call to adventure	True
□	Fact	plot point: action chosen	False

THEN

enable links to this node

when visited, update the following facts:

□	True/False	plot point: action chosen	True
□	True/False	end	True

Figure 2: Anywhere nodes and facts, listing nodes (upper-left) and facts (lower-left) in main window.

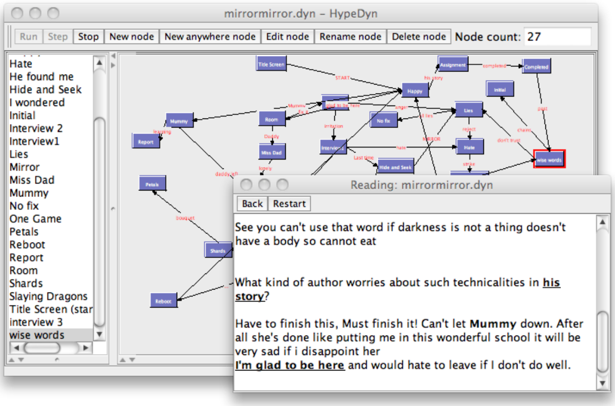
3. TEACHING WITH HYPEDYN

By allowing authors to create procedural hypertext through a visual interface, HypeDyn allows authors to explore issues of non-linear and adaptive interactive storytelling, and to confront issues such as choice, variability, coherence, recurrence and rereading without having to tackle the difficult technical issues which often arise when creating interactive stories.

We have been using HypeDyn as a teaching tool in our undergraduate class on interactive storytelling in the Department of Communications and New Media at the National University of Singapore. As part of their coursework, students create two short hypertext fictions: one which makes use of traditional approaches to hypertext, and one which uses more procedural approaches to create a sculptural hypertext. Both stories are required to consist of approximately 25 nodes, with a guideline of 100 words per node.

For the first story, students are asked to create a traditional node-and-link hypertext. The story, however, must make use of the procedural features of HypeDyn. Students are required to include at least two procedural changes to links, and two procedural changes to the text in the nodes. This exposes the students to the basic procedural elements of HypeDyn. As can be seen in the example in Figure 3, these stories consist of the usual network of nodes, with links between the nodes enabled or disabled based on the reader's previous actions. For example, the links "his story" and "I'm glad to be here" are currently active, whereas "Mummy" is locked.

Students' stories for this first assignment tended to follow two models. Many resembled the traditional model of hypertext fiction, with links embedded within paragraphs, and the meaning of the links often indirectly suggested by the linked words. Stories usually involved exploring a series of story fragments which, when taken as a whole, conveyed a narrative. The other common model was that of "choose-your-own-adventure" stories, where links represented choices as to "what will happen next" in the story.



mirrormirror.dyn – HypeDyn

Node count: 27

Hate
He found me
Hide and Seek
I wondered
Initial
Interview 2
Interview1
Lies
Mirror
Miss Dad
Mummy
No fix
One Game
Petals
Reboot
Report
Room
Shards
Slaying Dragons
Title Screen (star
interview 3
wise words

Reading: mirrormirror.dyn

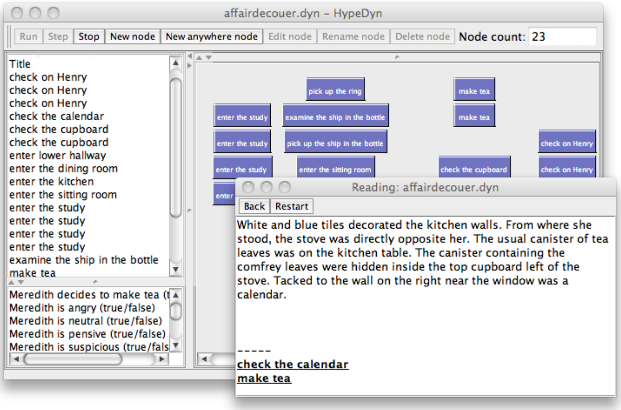
See you can't use that word if darkness is not a thing doesn't have a body so cannot eat

What kind of author worries about such technicalities in **his story**?

Have to finish this, Must finish it! Can't let **Mummy** down. After all she's done like putting me in this wonderful school it will be very sad if i disappoint her
I'm glad to be here and would hate to leave if I don't do well.

Figure 3: Example of traditional hypertext fiction, story written by Shane Kua.

For the second story, students are asked to create a sculptural hypertext. To enforce this requirement, they use a modified version of HypeDyn which does not allow for traditional links between nodes. Instead, they are only allowed to use "anywhere" nodes. Students are required to create at least three variations on the story, where a variation is defined as a path through the story where at least two nodes are encountered in a different order than in previous readings of the story. In addition, students were required to make use of facts to change when nodes can be seen, and to make use of facts to make procedural changes to the text within the nodes.



affairdecouer.dyn – HypeDyn

Node count: 23

Title
check on Henry
check on Henry
check on Henry
check the calendar
check the cupboard
check the cupboard
enter lower hallway
enter the dining room
enter the kitchen
enter the sitting room
enter the study
enter the study
enter the study
enter the study
examine the ship in the bottle
make tea

Meredith decides to make tea (t
Meredith is angry (true/false)
Meredith is neutral (true/false)
Meredith is pensive (true/false)
Meredith is suspicious (true/fals

Reading: affairdecouer.dyn

White and blue tiles decorated the kitchen walls. From where she stood, the stove was directly opposite her. The usual canister of tea leaves was on the kitchen table. The canister containing the comfrey leaves were hidden inside the top cupboard left of the stove. Tacked to the wall on the right near the window was a calendar.

-----
check the calendar
make tea

Figure 4: Example of sculptural hypertext fiction, story written by Carmen Pongdhana.

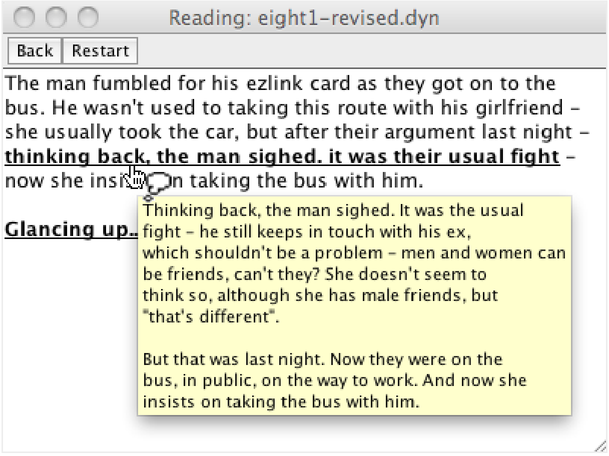
These requirements led to a very different type of story, where authors became more concerned with the preconditions which determined when nodes could be seen, rather than following a more traditional branching-path structure (see Figure 4). For this second story, students tended to create links which more explicitly indicated the meaning connected with selecting the link, frequently associating these links with actions taken by characters within the storyworld. They also used facts to keep track of, for example, the mental state of characters. These facts were then used to determine the result of selecting a link. For example, in the story shown in Figure 4, the names of the nodes listed in the upper-left of the main window tend to denote actions of the main character, Meredith, such as “check on Henry”, “check the cupboard”, or “enter the study”. Facts, listed in the lower-left of the main window, correspond to Meredith’s state of mind, such as “Meredith is pensive” or “Meredith is suspicious”.

It is interesting to note that many of the students’ sculptural hypertext stories appear similar to text-based interactive fiction (IF). It would be interesting to explore further whether introducing explicit representations of state (facts) to a hypertext authoring environment necessarily encourages an IF-like approach to storytelling, or whether other, different forms unique to sculptural hypertext will emerge.

Through these two projects, students were able to gain insight into theoretical issues related to authoring interactive stories, such as the problem of the explosion of content associated with branching stories, and the difficulties which authors face in terms of balancing authorial control and reader agency when creating more procedurally-driven stories.

4. USING HYPEDYN FOR RESEARCH

In addition to teaching, we have also used HypeDyn as a tool in our ongoing research into interactive storytelling. We have made use of HypeDyn to explore a number of different research topics related to interactive storytelling, including work to explore issues representation in authoring tools to support authoring of multiform stories [6], the limits of rereadability in procedural interactive stories [7], and the paradox of rereading in hypertext fiction [8].



The man fumbled for his ezlink card as they got on to the bus. He wasn't used to taking this route with his girlfriend – she usually took the car, but after their argument last night – **thinking back, the man sighed. it was their usual fight** – now she insi n taking the bus with him.

Glancing up..

Thinking back, the man sighed. It was the usual fight – he still keeps in touch with his ex, which shouldn't be a problem – men and women can be friends, can't they? She doesn't seem to think so, although she has male friends, but "that's different".

But that was last night. Now they were on the bus, in public, on the way to work. And now she insists on taking the bus with him.

Figure 5: Customized version of the HypeDyn reader with popup-links and context-sensitive cursors.

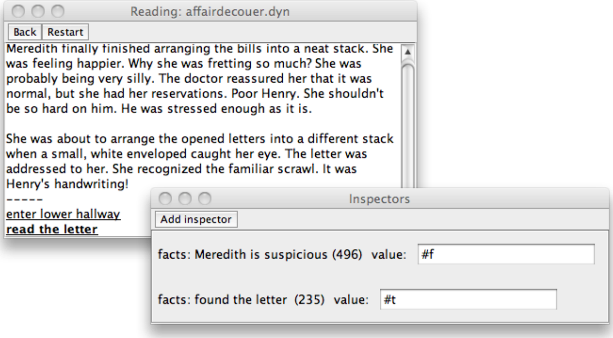
An example of the work which we have carried out using HypeDyn is our recent exploration of issues related to navigation and reader orientation within hypertext fiction. For this research, we wanted to explore how different ways of indicating the nature of links, and different types of transitions between nodes, impacted readers’ orientation within a hypertext story. In this project, we were able to quickly and easily make modifications to the HypeDyn reader interface to explore these ideas (see Figure 5).

In all of these studies, HypeDyn proved to be a flexible tool which could be adapted to suit the needs of the specific studies, while at the same time providing a stable platform from which to work.

5. LIMITATIONS

Although HypeDyn has proven to be an effective teaching and research tool, there are clearly a number of ways in which it could be improved. These include problems which arise when using traditional hypertext representations for authoring procedurally-driven stories, limitations of primitive node-and-link constructs when authoring larger hypertext fictions, and issues of procedural expressiveness versus the need to provide a tool which is accessible to non-programmers.

One problem which became clear in the course of the second project in our interactive storytelling class was the limitations of the traditional hypertext map visualization when a hypertext becomes highly procedural. Much of the structure in a procedural hypertext is created through the use of facts and their associated rules. In a node-and-link map view, this procedural structure is not visible to the author. What is still dominating the representation is the pre-authored link structure, a structure which may not actually even be present in a completely procedurally-driven story. This suggests the need to explore new, alternative visualizations and representations, which may either complement or completely replace the existing map view.



Meredith finally finished arranging the bills into a neat stack. She was feeling happier. Why she was fretting so much? She was probably being very silly. The doctor reassured her that it was normal, but she had her reservations. Poor Henry. She shouldn't be so hard on him. He was stressed enough as it is.

She was about to arrange the opened letters into a different stack when a small, white enveloped caught her eye. The letter was addressed to her. She recognized the familiar scrawl. It was Henry's handwriting!

-----

enter lower hallway

read the letter

facts: Meredith is suspicious (496) value: #f

facts: found the letter (235) value: #t

Figure 6: Debugging tools in HypeDyn.

In addition, the extensive use of facts and rules introduces the need for richer debugging tools. Currently, HypeDyn provides a simple “inspector” interface (see Figure 6), which enables authors to see the state of nodes, links and facts at runtime. However, this interface is very limiting, and does not directly represent the current state of the paths available in the story. For an author from a non-technical background, it can be very hard to make the connection between a list of facts and the current state of the story. Insight into how this work should proceed can be drawn from other interactive storytelling systems. For example, the Inform

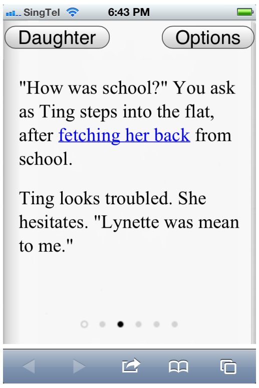
7 interactive fiction authoring system provides support for recording and replaying of different paths through a story, which is visualized as a branching tree or "skein" [9]. A similar approach may be appropriate for procedural hypertext fiction.

We have also started to encounter limitations in terms of the primitive node-and-link constructs when authoring larger works. For example, when creating a moderately complex hypertext fiction containing 69 nodes and 205 links for a research project [8], we already began to encounter difficulties visualizing the overall structure of the story. This is a problem which other researchers have also addressed, with some higher-level structures such as "hypersections" [10] having been suggested.

A third limitation which we have observed with the current design of HypeDyn is the lack of procedural expressiveness. Our students, who are generally not programmers, began running up against the limits of the simple if/else constructs provided in the HypeDyn rule interface. To work around this, they tended to create multiple copies of nodes. This allowed them to indirectly create if/elseif/else structures, but created problems in terms of duplicate content. One approach to provide more expressiveness would be to allow for a more general rule structure which moves beyond a simple if/else construct. We have begun to experiment with this in the latest version of HypeDyn. The challenge is how to maintain the simplicity of the system, while at the same time providing for greater procedural expressiveness.

6. STATUS AND FUTURE DIRECTIONS

HypeDyn has been implemented in Kawa [4], and has been tested on MacOS, Windows, and Linux. HypeDyn was released as an open-source system in December 2011 (see http://www.partechgroup.org/hypedyn/). Stories can currently be published either to a standalone Java application, or as a Java applet embedded in a web page. We would like to make the finished stories more easily accessible across various platforms, in the form of an HTML5 reader. We have recently released a prototype version of this new reader which runs on both web and mobile platforms (see Figure 7), opening up the possibility of exploring cross-platform, embodied, and location-based storytelling.



"How was school?" You ask as Ting steps into the flat, after fetching her back from school.

Ting looks troubled. She hesitates. "Lynette was mean to me."

Figure 7: Prototype HTML5 reader; story written by Sameul Heng, prototype reader implemented by Teong Leong Chuah.

HypeDyn has been used as part of our research and teaching, and we intend to continue to develop the system as part of our ongoing work. Our immediate plans are to provide more procedural expressiveness while continuing to support non-programmers. This will involve providing different representations and debugging tools for the more procedural aspects of the system as the underlying abstractions move further away from the predefined node/link structure of hypertext. We also plan to explore different reading interfaces, explore story-specific abstractions, and provide higher-level constructs such as patterns [1] to support fiction authoring.

7. CONCLUSION

Our experience with HypeDyn, both in teaching and in research, has shown that it is possible to explore issues in interactive storytelling through the use of hypertext fiction. We believe that it is possible to push the bounds of what is traditionally considered "hypertext", exploring new forms of procedural expressiveness and dynamic, computationally-driven interactive storytelling, while still retaining the conceptual simplicity of hypertext which makes it accessible to non-technical authors and students. We are keen to see how people will use and modify HypeDyn to suit their needs.

8. ACKNOWLEDGMENTS

This work was funded under a Singapore-MIT GAMBIT Game Lab research grant, "Tools for Telling: How Game Development Systems Shape Interactive Storytelling."

9. REFERENCES

[1] M. Bernstein. Patterns of hypertext. In *Proceedings of Hypertext '98*, pages 21–29. ACM Press, 1998.

[2] M. Bernstein. Card Shark and Thespis: exotic tools for hypertext narrative. In *Proceedings of Hypertext '01*, pages 41–50. ACM Press, 2001.

[3] M. Bernstein. Storyspace 1. In *Proceedings of Hypertext '02*, pages 172–181. ACM, 2002.

[4] P. Bothner. Kawa: compiling dynamic languages to the java vm. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 41–41. USENIX Association, 1998.

[5] R. Kendall and J.-H. Réty. Toward an organic hypertext. In *Proceedings of Hypertext '00*, pages 161–170, New York, NY, USA, 2000. ACM.

[6] A. Mitchell and K. McGee. Designing hypertext tools to facilitate authoring multiple points-of-view stories. In *Proceedings of Hypertext '09*, pages 309–316. ACM Press, 2009.

[7] A. Mitchell and K. McGee. Limits of rereadability in procedural interactive stories. In *Proceedings of CHI '11*, pages 1939–1948. ACM Press, 2011.

[8] A. Mitchell and K. McGee. The paradox of rereading in hypertext fiction. In *Proceedings of Hypertext '12.* ACM Press, 2012.

[9] G. Nelson. Natural language, semantic analysis, and interactive fiction. In K. Jackson-Mead and J. R. Wheeler, editors, *IF Theory Reader*, pages 141–188. Transcript On Press, Boston, MA, 2011.

[10] J.-H. Réty, S. Bouchardon, J. Clément, N. Szilas, and C. Angé. Rencontre : an experimental tool for electronic literature. In *Proceedings of Electronic Literature in Europe*, 2008.