Hyperstories for Learning

Jaime SANCHEZ 1, Mauricio LUMBRERAS 2, Luis Mariano BIBBO 2

1 Department of Computer Science
University of Chile
Santiago, Chile
Email: jsanchez@dcc.uchile.cl

2 LIFIA - Laboratorio de Investigación y Formación en Informática Avanzada, Universidad Nacional de La Plata
Calle 50 y 115, 1900, La Plata, Argentina
Email: [mauricio, lmbibbo]@info.unlp.edu.ar

ABSTRACT

We present a conceptual model for building hyperstories. Hyperstories are the hypermedia version of literary stories. The model combines static and dynamic aspects of a computer environment such as a nested context, allowing navigation through a virtual world. The flexibility of the model supports things such as the existence of objects to be acted on by the learner, autonomous objects or characters who represent entities that live independently from users, the reusability of entities and environments to avoid repetitive work, and a clear separation between content representation and interface management .

We introduce a methodology for using hyperstories to enhance learning and thinking in a constructivist way. We discuss different aspects involved in the implementation of hyperstories. Finally, we analyze some further trends and issues in this growing line of research.

Keywords: Educational hypermedia, dynamics objects, modeling technique, edutaintment

Hyperstories for Learning

Jaime SANCHEZ 2, Mauricio LUMBRERAS 2 , Luis Mariano BIBBO 2

1 Department of Computer Science
University of Chile
Santiago, Chile
Email: jsanchez@dcc.uchile.cl

2 LIFIA - Laboratorio de Investigación y Formación en Informática Avanzada, Universidad Nacional de La Plata
Calle 50 y 115, 1900, La Plata, Argentina
Email: [mauricio, lmbibbo]@info.unlp.edu.ar

ABSTRACT

We present a conceptual model for building hyperstories. Hyperstories are the hypermedia version of literary stories. The model combines static and dynamic aspects of a computer environment such as a nested context, allowing navigation through a virtual world. The flexibility of the model supports things such as the existence of objects to be acted on by the learner, autonomous objects or characters who represent entities that live independently from users, the reusability of entities and environments to avoid repetitive work, and a clear separation between content representation and interface management.

We introduce a methodology for using hyperstories to enhance learning and thinking in a constructivist way. We discuss different aspects involved in the implementation of hyperstories. Finally, we analyze some further trends and issues in this growing line of research

1. CONCEPTUALIZING HYPERSTORIES

Conceptually we define a hyperstory as:

HS = 1. Hypermedia + 2. Dynamic objects + 3. Character

1. Is used to represent the environments and links among them.
2. Are entities that have certain behavior in time and react to the events produced by users and other entities.

3. Is a special entity called protagonist. This is manipulated by users and represents the connection with the system.

A hyperstory is seen as the electronic version of a conventional literary story, in the same way as hypertext is the electronic version of text. However, we have improved this analogy by allowing a "dynamic binding" among characters, the world in which they move and the objects they act on[Sánchez 95]. This binding is performed by the learner, thus allowing greater flexibility in the learning process.

In other words, a hyperstory is a combination of a virtual world where the learner can navigate, a set of objects operated by the learner, and a group of characters manipulated by the learner[Sánchez 94]. Objects and characters may have their own behavior and act autonomously. Learners (when manipulating a character) can also interact with other characters to solve a given problem. Familiar environments such as schools, neighborhoods, squares, parks, supermarkets, and the like, could be interesting metaphors for building virtual worlds. Hyperstories are navigated through the use of a hypertext-like metaphor and can be good examples of powerful multimedia tools based on constructivist epistemology.

2. RATIONALE

Traditional models of educational software engineering are based on the tutorial mode of instruction by focusing on book-like presentation design guidelines and stemming from a cycle of presentation of learning content-questioning-presentation-questioning and so on [Bork 91]. A much more flexible model of educational software is the educational game that incorporates key cognitive strategies motivating and fully involving learners by giving them control over the learning task, challenging, engaging, interacting, and adapting to the player's level, ranging from beginning to advanced [Kafai 94, Allesi 91].

This study aims at improving thinking and learning through the use of hyperstories. Stories are narratives of true or fiction events that intend to capture and involve learners actively. Hyperstories were built to enhance the development of cognitive structures that determine tempo-spatial relationship and laterality in early age children. Children without an adequate development of these intellectual structures, cannot be aware of either their own position in time and space, the position of others, the position of others in relation to them, the position of the things and objects that surround them, and the recognition of their and other's left and right [Sánchez 93]. Many authors believe that if these cognitive constrains are not treated early, the mental profile of border line learners may decrease.

We build hyperstories in order to engage, challenge, interact, and motivate learners by giving them control over stories, tools, and construction materials to build things and develop strategies to test hypothesis with the implicit idea of fostering the development and use of tempo-spatial relationship and laterality. By playing with hyperstories we intend to exploit the notion of space, position, sequence, and extension in time, as well as enhancing the development of higher order cognitive structures.

As a consequence, we believe that hyperstories can contribute to assist learners in the development and early structuration of tempo-spatial relationships and laterality. Children like stories and remember them easily. They tend to exploit their imagination. When a child gets engaged in a story, s/he can identify, retrieve and use relevant data to solve a challenge by having rapid and flexible access to the story sequence. Hyperstories motivate learners, facilitate navigation, and promote long-term learning by providing a rich context of information and making it more memorable.

3. CONCEIVING A CONCEPTUAL MODEL

Conventional hypermedia authoring tools do not provide an adequate set of facilities for building hyperstories. The static environments involved in a hyperstory (virtual world) can be simulated easily, but several aspects such as the behavior of dynamic objects and complex interactions between the main character and others exceed the conventional "nodes and links" model, as seen in [Halasz 90, Furuta 90, Lange 90].

What are our requirements in order to conceptualize this model?

- Isolation of the interface from the content of the story
- Composition, modularity, and inheritance among entities.
- Support of concurrent events.
- Independence between the specification and the implementation language.
- The user interface implemented in a different language from the implementation of the story
- Objects with dynamic and autonomous behavior.
- Synchronous and asynchronous communication among entities.

Our specially defined model tries to provide tools in order to satisfy these requirements. By using a specially defined conceptual model for building hyperstories we can describe the virtual world as a kind of nested context model. The world can be enriched with objects and characters and thus "instantiating" a particular story.

A virtual world is defined as a set of contexts that represents an environment. Each context contains an internal state, a set of contained contexts, a set of objects, links to other contexts, and a specific behavior. Different relationships may be held between two different contexts, such as:

- neighborhood (there is a link from one context to the other),
- inclusion (one context is included in the other),
- none (contexts are "disjoints").

The idea of a context is the same as in standard hypertext technology: a node or container. Different "real world" metaphors can be implemented easily with this simple model, such as a town, a house, a room (or houses within a town and rooms in a house), or a book . All these metaphors are built in such a way that can be browsed and navigated. The main difference between our model and traditional hypermedia models is that nodes (contexts) may be nested, in some way like [Casanova 91].

State variables and behavior describe the attributes of a given object. The object's behavior is specified by using rule-based scripts. Each rule contains a triggered event, a pre-condition, and a list of actions that must be performed when the event arrives and the pre-condition holds. An object can behave differently in its life stage for the same message received, in some way like [Tailvasaari 93]. In a way these rules are not specified in a flat way, they are blocked and grouped according the entity life stage. According to their behavior, objects can present static or dynamic characteristics. For example, a static object always belongs to the same context (e.g., a door, a window, etc.) and a dynamic object may be carried from one context to another or may have certain dynamic activities. For instance, the learner may pick up a key from a table and keep it while s/he is navigating through the world.

It is important to state that certain contexts, for instance books, can be also carried from one container context to another. We call them dynamic contexts. This capability must be expressed in the behavior of this context. In addition, objects can perform discrete or continuous activities. A door can be opened or closed (discrete) and a recipient may be filled with water (continuous). Finally, one of the most interesting characteristics of certain objects is the capability of representing links between contexts(e.g., a door communicates two different rooms). Characters allow the learner to communicate with the virtual world. When manipulating characters the learner acts upon the world, navigates, performs actions on and with objects, carries objects from one context to another, etc.

Considering what was mentioned above, a hyperstory is then the combination of contexts, objects, characters, and the interaction patterns performed by the learner. Events produced by the learner while interacting with the virtual world, the time sequences, and the interaction among objects, determine a particular "instantiation" of a hyperstory. Two different learners may experience different views of the same virtual world. Slight changes introduced by the learner to the object's behavior will produce different hyperstories in the same world. As a consequence, this model allows the use of the same virtual world for different learning purposes.(e.g.: modifying the interface (not the content), you can build a story presenting all the information using sound, in case you want to build a hyperstory for a blind child [Lumbreras 93])

In addition, the model allows us to build new contexts by getting profit from old ones (e.g., a standard house can be inherited by an expensive house by including a pool. In this way, the expensive house has all the attributes of the standard house and a new subcontext). This concept is mapped to all kind of entities (objects, characters, links, etc.)

In relation to concurrence, it has to be taken into account that certain activities could be performed out of the user perception(e.g. when you open a faucet and you come back later, water begins to fill a bucket. If you leave the room, the bucket will be full of water). The system allows us to model concurrent activities by specifying special events managed by the timer.

4. WRITING AND MODELING A HYPERSTORY

When you write a hyperstory, the first idea is to write a flow chart or state diagram [Kelso 93] that describes the different paths of the story, and the main events that can modify it. This is useful in the first stage. The method can easily produce a monolithic piece of code, very difficult to maintain and debug. A more natural way is to think about the behavior of each object or character, and to map the main events that can change the story to rules, like writing a script for each drama actor. These rules describes the object's behavior. As a result we have fragmented the flow chart into each object. Therefore maintenance and debugging is easy, because the cause-effect rule is contained in the object behavior.

4.1. Extending the navigation space

We note that there are two hyperspaces running simultaneously:

- The static environment that support the navigation,
- The story space.

In the last one, there are links and nodes, but not in the standard way, because one node could be conceptualized such us a determined state of the instance variables, behavior of the objects, and characters. These instance variables or the behavior can change due to certain events generated by the user or some object. A subset of these events can change the course of the story In this case a link have been activated. Later in the "vampire" example we show these concepts.

Our model pushes forward the hyper structure. By combining certain non-deterministic rules and timing programming, that can be triggered in several system states, you can assign certain autonomous behavior to objects. In that way you avoid redundant specification, because these rule behaves like a demon.

4.2. The vampire example

Suppose that there is a vampire that travels autonomously for several contexts in a castle. The behavior of the vampire is to verify if there is a person in the context, and then he will try to bite him/her. For example, the victim become transformed in a vampire when is bitten. To do this, the victim needs a rule that enables this act.

We cannot know in advance which physical context or when the vampire could bite somebody. In this case the behavior represents a link that goes from several story states (story nodes) to others. In the destination node, the character will have the changes made by the vampire.

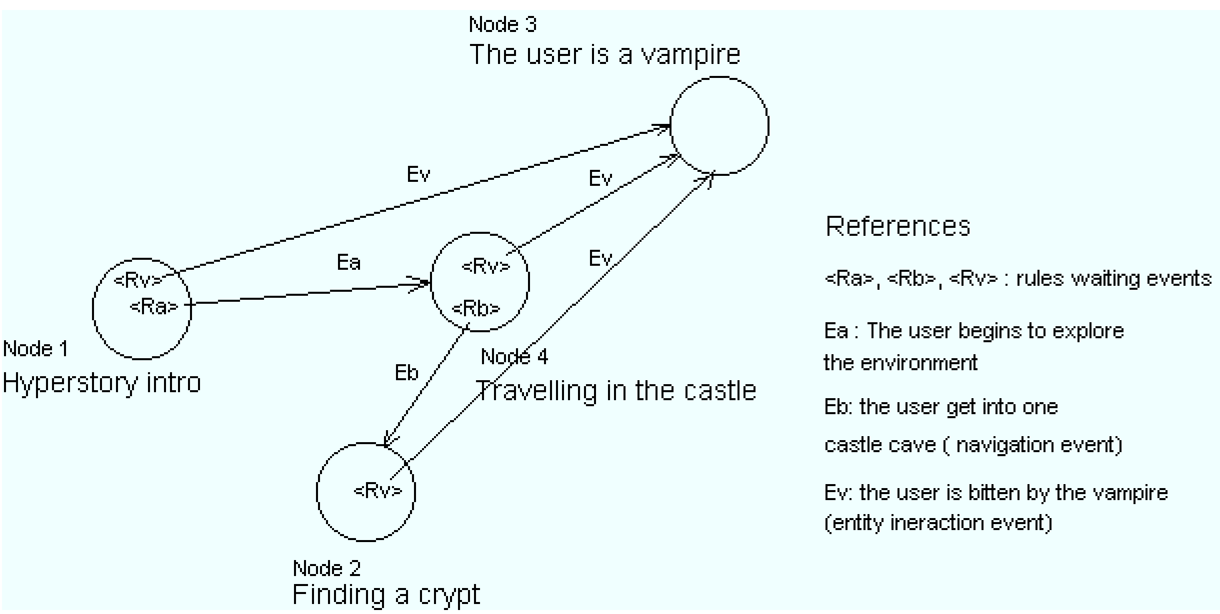

Figure 1: Abstract representation of a hyperstory excerpt

Nodes represents story phases. Rules can be viewed as anchors (<Ra>, <Rb>, <Rv>). Events trigger anchors (Ea, Eb, Ev), and generate navigation in the hyperstory space. Conceptually, demons rules act as anchors present in several nodes, but the model allows to specify it only once(in some way like a button on the background in a Hypercard application).

5. IMPLEMENTING HYPERSTORIES

Our model presents a special notation in order to describe each stage of the conceptual world design, providing four special constructors:

- context: to represent an environment
- link: to relate two contexts allowing navigation
- entity: to represent an object or a character
- channel: to connect explicitly the events among entities

Each constructor defines one class, that can be redefined and instantiated any number of times. Some guidelines to compose a hyperstory are:

- Describing the static world. The description of the global navigation structure. The task involves the specification of each environment (context) and the linkage and nesting among them. In this stage we use two constructors: context and link.
- Describing objects. The detailed description of each behavioring entity or relevant attributes. We use the entity as constructor in this stage .

- World + object composition. The filling of each environment with adequate entities. Apart from this, the designer needs to associate each conceptual entity with the adequate interface element.

One important thing to take into account in the whole development process, is that we can use some old definition (taken from another hyperstory) in order to reuse or redefine a new entity. In this case you have a powerful tool (e.g., you can build another story using an old environment, or you can define a coffee machine reusing and inheriting from a standard electric device previously defined).

6. SYSTEM ARCHITECTURE

Basically the system is composed in some way like [Appino 92, Lewis 91], by:

- The kernel,
- The interface manager

The kernel maps every interface object (such us characters, objects, contexts, etc.) to an internal object. The kernel's object can send messages to the equivalent interface object in the interface. The communication between internal and interface objects is made through a special language and protocol.

The interface has the capability to receive and show the incoming messages sent by the kernel. Furthermore, the interface manager captures the user activity, and dispatches it to the kernel.

As stated previously, conventional hypermedia authoring tools reflects a lack of facilities to implement hyperstories requirements. Our architecture solves this drawback, because the timing and complex object behaviors (managed by the kernel) are implemented by using an adequate language that manages this constrain. Also, there are several tools that provide an easy user interface management. In that way we made the system to profit from the best tools available. Different interfaces and networked hyperstories could be easily implemented with this global design.

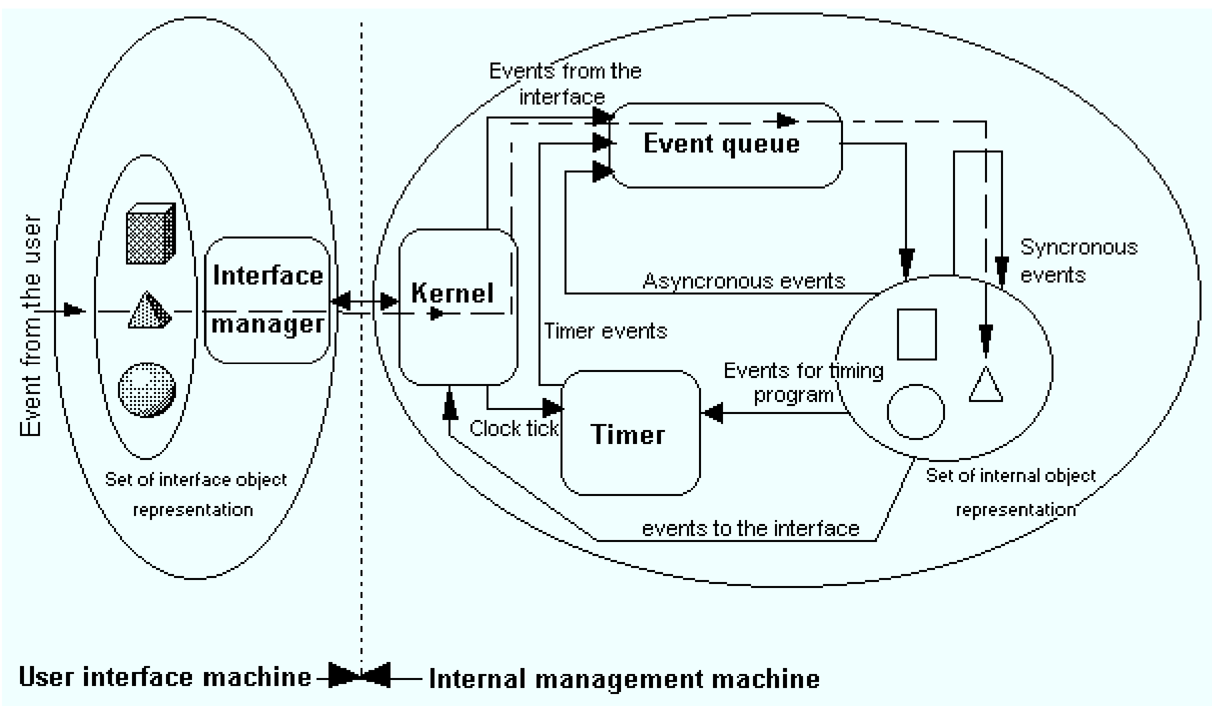

Figure 2: Graphical representation of the system architecture'

7. NEW IDEAS, NEW PROBLEMS

As previously stated current hypermedia authoring environment are not ideal target environments for hyperstories. Once designed, a hyperstory involves not only a static world (usual in the hypermedia field) but also a set of dynamic entities (objects, characters) that must be synchronized in order to perform the hyperstory. Conventional hypermedia environments can only support a restricted version of the static aspects of our model. The ideal "hyperstory's engine" is the combination of a hypermedia engine(that deals with navigation around the virtual world) with a concurrent object-based architecture (that serves objects and characters and dispatches internal and external events).

Our first prototype is built around Multimedia Toolbook V3.0 (that manages everything related to the interface management) and through Windows message passing mechanism, the interface sends and receives information from a kernel written in Turbo Pascal for Windows using OOD concepts. Each conceptual entity of the system is described in a written code. We associate each conceptual entity with the adequate interface element. The first prototype shows that the complex aspects are the performance of animation rendering within MS-Windows and the need for giving to the system designer a browser to build the application easily .

8. CONCLUDING REMARKS

We have presented a conceptual model for building hyperstories. It has several advantages with respect to conventional hypermedia data models: for example it is based on a nested context model where each context may represent a physical place (a house, a room) or a logical one (a complete hyper environment). Furthermore, it includes static and dynamic objects (that stand for physical or conceptual entities in the virtual world) and support complex interactions between the learner and the world.

We are now working on several directions to improve our model. First, we are building prototypes supporting a sub-set of the model's features. We are also building experimental hyperstories using conventional hypermedia authoring tools(e.g., Toolbook) for testing our ideas about the use of hyperstories as educational tools. We intend to define different interface styles in order to adapt hyperstories for learners with special needs. In addition, our research group is studying some issues related to the process of building hyperstories and the type of tools needed to do this.

Building a hyperstory is obviously more complex than building conventional hypermedia-based educational material. Traditional hypermedia modeling techniques[Garzoto 91] must be extended in order to support dynamic aspects of hyperstories. Moreover, as it was repeatedly stated high level tools are needed if we want to simplify the construction of hyperstories. We feel that if we solve this problem we will gain knowledge about the rhetoric of hyperstories thus empowering our software environments.

REFERENCES

[Allesi 91] S. Allesi, Trollip, S. " Computer-based instruction: Methods and development", New Jersey: Prentice Hall Inc., 1991.

[Appino 92] Appino P., Lewis J., Koved L et al, "An Architecture for Virtual Worlds", Presence, Volume 1, Number 1, Winter 1992, MIT Journals.

[Bork 85] A. Bork, "Personal computer for education", New York: Harper & Row Publisher, 1985.

[Casanova 91] M. Casanova, L. Tucherman, J.L. Neto, N. Rodriguez, L.Soares, "The nested context model for hyperdocuments", Proceedings of Hypertext '91.

[Furuta 90] Furuta R., Stotts D., "The Trellis Hypertext Reference Model", In Proceedings of the Hypertext Standarization Workshop, National Institute of Standards and Technology, January 1990, pp. 83-93.

[Garzoto 91] F. Garzotto, P. Paolini, D. Schwabe, "HDM. A Model for the design of Hypertext applications". Proceedings of Hypertext '91.

[Halasz 90] Halasz F., Schwartz M., "The Dexter Hypertext Reference Model", In Proceedings of the Hypertext Standarization Workshop, National Institute of Standards and Technology, January 1990, pp. 95-133.

[Kafai 94] Y. Kafai, E. Solloway, "Computational gifts for the barney generation", Communications of the ACM, 37(9), pp.19-22., 1994.

[Kelso 93] M. Kelso, P. Weyhrauch, J. Bates, "Dramatic Presence", Presence, Volume 2, Number 1, Winter 1993, MIT press.

[Lange 90] Lange D., "A Formal Model of Hypertext", In Proceedings of the Hypertext Standarization Workshop, National Institute of Standards and Technology, January 1990, pp. 145-166.

[Lewis 91] Lewis J., Koved L. & Ling D., "Dialogue structures for virtual worlds", In CHI'91 Conference Proceedings, May 1991, New Orleans, USA. pp. 131-136.

[Lumbreras 93] M. Lumbreras, "A hypertext for blind people", poster of Hypertext '93, Seattle, 1993.

[Sánchez 93] J. Sanchez, "The conceptualization of hyperstories", Working Document, 1992.

[Sánchez 94] J. Sánchez, Mallegas, A., Cernuzzi, L., Rossi, G., Lumbreras, M. And Díaz, A. "A conceptual framework for nuilding hyperstories". In Ed-Media '94 Conference Proceedings, p.761.

[Sánchez 95] Sánchez, J. & Lumbreras, M., "Interfaces for learning". To be Published in Advances in Human/Factors: Human-Computer Interaction. Elsevier Publishers.

[Taivalsaari 93] Taivalsaari, A., "Object-Oriented programming with modes", pp. 25-32, June 1993.