PastMaster@Storytelling: A Controlled Interface for Interactive Drama

Nicolas Szilas
Macquarie University
Department of Computing
Macquarie University NSW 2109
+61 2 9850 9571
nicolas@ics.mq.edu.au

Manolya Kavakli
Macquarie University
Department of Computing
Macquarie University NSW 2109
+61 2 9850 9572
manolya@ics.mq.edu.au

ABSTRACT

In this paper, we describe a controlled interface, *PastMaster@Storytelling*, for Interactive Drama. It is developed for a player to interact with an interactive drama engine. The proposed interface has been tested and the results are presented.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – Navigation, User issues. I.2 [**Artificial Intelligence**]: *games.* J.5 [**Arts and Humanities**]: *Linguistics, Literature, Performing arts.*

General Terms

Design, Human Factors.

Keywords

Interactive Drama, Interactive Narrative, Adaptive Interfaces, Interaction History, Narrative Intelligence.

1. INTERACTIVE DRAMA

As a medium, the computer can provide novel ways to tell stories with the active participation of the audience (e.g. [7, 11]). Interactive Drama (ID) is an experience, where the audience acts as a character by making decisions on each character's actions in a story. Interactive Drama has become an active and challenging research area involving various branches in human and computer sciences such as Intelligent Agents, Narrative Intelligence, Human Computer Interaction (HCI) and Language Technologies.

Narrative and story are two terms confused and misused. In this study, the story refers to the sequence of chronological events occurring in the fictional word and the narrative refers to the sequence of events as they are presented to the audience [5]. The term drama is used to denote a specific type of narrative, in which the audience directly perceives the characters' actions, rather than being told the events.

The specificity of ID compared to other forms of digital storytelling is that the audience has the possibility to modify not only the narrative but also the story itself. The key issue is to depart from scripted stories described as storygraphs because this kind of solutions strongly limits the interactivity. Generative algorithms are needed to provide agency to the user, that is "the satisfying power to take meaningful action and see the results of our decisions and choices" [11 p. 126]. Those algorithms are being investigated by various research groups [2, 15, 16, 21, 23], yielding numerous prototypes.

The system developed by the authors, called IDtension is such a prototype [16, 17]. It allows the user to play a main character in a story by choosing his action. For example, the user can transmit information to other characters, influence them, ask them assistance to perform some tasks, blame them to have performed some tasks, etc. The narrative engine of IDtension allows to calculate in real time the actions of non player characters, according to narrative constraints. These narrative constraints are implemented via a Model of the User, which estimates the impact of each possible action on the user according to several narrative criteria [17]

The output of the engine is text: actions are displayed through a template-based language generation system. For example, the user would read on the screen: "Mary to you: you should steal the key from Allan". Although we have been working recently on the integration of the narrative engine to a 3D world, with a game engine called *Unreal Tournament*, the design and experiments that are reported in this paper are based on the text version.

The input of the system consists in choosing actions for the character that the user is controlling. Departing from a mere list of possible choices, we have designed a method to interact with the system, called *PastMaster@Storytelling*. *PastMaster* is based on the interaction history.

The rest of the paper is organized as follows. Section 2 exposes the issue of action selection in Interactive Drama and the two alternatives of solutions to resolve the multiple action selection. Section 3 overviews the existing solutions belonging to the group of controlled interfaces. Section 4 presents a new approach for action selection and Section 5 provides a preliminary evaluation of this interface. Section 6 extends the approach to a wider set of applications.

2.INTERACTION DESIGN OF INTERACTIVE DRAMA

2.1 The Choice Problem

The visible difference between ID and other forms of interactive narrative (hypertext, Interactive Fiction, adventure video games, etc.) is the number of narrative actions that the user can undertake (the range of actions, to use Brenda Laurel's terminology [7 p. 20]). In an adventure video game for example, only a few actions have a significant effect on the story (usually, only one action makes the story go forward, others are "fails"). In our system,

Conference'04, Month 1–2, 2004, City, State, Country.

considering only one type of dialog act, such as “ask for assistance”, one may suppose that the user’s character has to perform 5 tasks and assuming that there are 5 other characters, it makes 25 possible “ask for assistance” acts. There are many types of acts, and we could observe experimentally the quickly growing number of choices given to the user [17]. Finally, the user ends with dozens of actions to choose from. Having them choose from a mere list is not acceptable, in terms of cognitive load. We denote this usability problem the *Choice Problem* [18]. Note that in a 3D world, the choice of physical acts is made easier by the environmental limitations, which is not the case for dialog acts. In the rest of the paper, our focus will be mostly on dialog acts.

The “Choice Problem” can be seen as the other side of the coin of agency in ID. Agency requires freedom and freedom means a lot of choices. The Choice Problem is similar to the problem of designing an interface for a complex system. There are however two specific important points to consider:

- It is not expected that the user will take time to learn how to use a complex user interface, because the target is entertainment.
- The interface should not be overwhelming for the user, who has to be immersed in the story. Ideally, the interface should be as transparent as possible.

2.2 Free vs controlled interfaces

In the field of Interactive Drama there are two main groups of solutions to allow the user to select an action.

In free interfaces, the user interacts with the system by using natural language. Free interfaces use either speech [2] or typed text [10]. Free interfaces enable the user to naturally dialog with other characters, with a transparent interface. There are two major problems [18] regarding free interfaces:

- It is technically not achievable to fully understand natural speech or text. Compromise should be found to achieve a certain level of understanding, or at least provide the user the feeling that s/he is understood.
- The narrative engine itself can only interpret a limited number of user's actions. Even if this number is large, say tens of actions, it is largely exceeded by the diversity offered by the free interfaces. As a result the free interface create false expectations, which must be "solved" by intelligently avoiding to interpret the user's sentence (for example suddenly changing the topic of the talk).

In controlled interfaces [18], the user can choose explicitly among a set of actions which contains not more and not less than the total number of available actions, as calculated by the narrative engine. Typically, a direct interface would use a classical menu interface to navigate through the actions.

Making explicitly available a large amount of choices does not solve the Choice Problem per se. A proper design has to be achieved in order to make the choice usable.

The controlled interfaces raise two issues:

- There is a risk that the resulting user interface is too overwhelming, disabling the immersion within the story, even in a non 3D representation.
- Making all choices explicit tends to put every narrative possibility at the same level (for example "Congratulate Mary" and "Insult Mary"), which is not suitable from a narrative point of view [10].

Current prototypes of Interactive Drama are using either free interfaces [2, 15] or controlled interfaces [3, 17]. Our goal is not enter into a debate about which one is better than the other, because it seems obvious that no ideal solution has emerged so far. Our approach focuses on the controlled interfaces, and proposes an innovative solution to overcome its limitations.

3. CONTROLLED INTERFACES FOR INTERACTIVE DRAMA AND GAMES

3.1 Selection in a list

The obvious and simplest way to provide a set of actions is to organize them within a choice list.

This solution is usual in adventure video games, for dialog interaction. The player is presented a short set of possible answers during the interaction with another character.

However, when the number of choices exceeds a certain size (typically seven), the classical problem of cognitive overload becomes an issue, considering the short term memory limits of human information processing.

Different combinations can be produced transforming the list into a map, categorizing the actions into groups. However, when one hundred or more actions are available, this is not a suitable solution either.

Other solutions should be investigated, which involve a hierarchical organization of actions into tree-structures.

3.2 Sentence building

Another approach consists of following the syntactical structure of each dialog action. An action is built as a sentence, starting with the subject (usually the user's character by default), the verbs and the various complements.

For example the *WYSIWYM* project from the University of Brighton [22] project is meant to allow users to easily enter information into a database. To do so, the user is presented a sentence with abstract terms to click on. When the user clicks on one of them, a list of choices is presented, and the phrase is updated. Step by step the user builds his/her own sentence, which corresponds to a knowledge representation in the system. For example, to create the sentence "Print the Name Field", the following interactive sentences are successively generated:

> [some action]
> Print [something]
> Print the [some label] field
> Print the Name field .

Thus, a dynamic user interface is generated automatically, according to the structure of the knowledge base.

A similar approach is currently under development in the field of Interactive Drama in the *Erasmatron* project, but in a more graphical manner [4]. *Erasmatron* is a long term project on Interactive Drama [3]. Its new interface consists in building a sentence incrementally. For example, to express "I flatter Mary a great deal", the user would follow a four-step procedure, as depicted in Figure 1.

Only a few video games have used this kind of interfaces, due to limitation of the set of available actions. One example is the game *Sentient* for PC, which allowed a very wide choice of actions (see Figure 2).

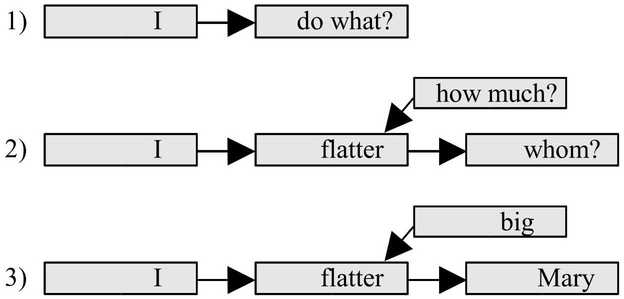


Figure 1. Example of a sentence-based interface for Interactive Drama: The Erasmatron project.

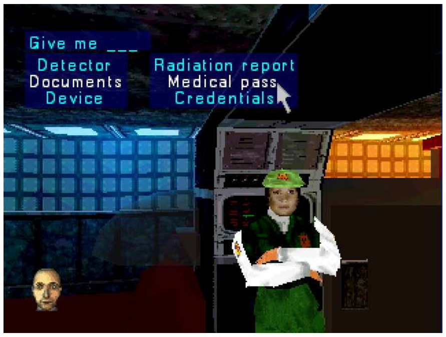


Figure 2. Sentence-based interface, in the game Sentient.

A variant of the sentence-based approach consists in using a controlled language rather than a menu based interface. A controlled language allows the user to type a sentence via the keyboard. But only sentences that fit to a given formal grammar are accepted. The user is thus guided to write only sentences that are understood by the computer [14].

The sentence-based interfaces allow the construction of structured action of arbitrary complexity. The progressive construction of the sentence guides the user during the action construction, by providing a naturally readable feedback. However, because they strictly follow the grammatical structure they are difficult to use, especially in an entertainment context. The number of clicks or keystrokes is also high.

Furthermore, this approach makes explicit the underlying structure of the actions, which is not useful in the case of Interactive Drama. In the context of the engine developed by the authors (*IDtension*), an action like "inform Clara that I could not start a riot because the gate is closed" requires the manipulations of notions like tasks, hindering and obstacles, which are not meant to be so visible.

In short, sentence-based interfaces are powerful but tend to hinder the immersion into the story, because of their complexity.

3.3 Object-based menus

Another way to construct a large set of actions is based on the objects that those actions manipulate. For example, to listen to the radio, the user clicks on the radio and a contextual menu pops up to provide various alternatives, including "listen to".

This approach is widely used in the video games. In the Game *The Sims* for example clicking on a character provides a list of several potential actions such as "Talk", "Joke", "Tickle, "Compliment", "Insult".

Since the user's attention remains on the fictional world, this approach support the feeling of immersion, but is not suited to dialog acts for the following reasons:

- Dialog acts are complex, involving several possibly hierarchical components
- Dialog acts involve objects that are not present in the current scene. This problem is easily be solved with an inventory, that is the list of objects possessed by the user's character. However, some objects are not in the user's possession. In addition Dialog acts involve abstract notions. The notion of object has to be extended to abstract notions such as "Mary has the key", or even "Mary did not manage to go to the yellow house because the door was closed".

Object-based menus make use of an interesting feature, adaptability. Indeed, the list of actions presented from a given object varies according to the current context. In the game *The Sims,* for example, the action "Kiss" would be proposed only if certain conditions are met, typically not at the beginning of the game.

4. *PastMaster* APPROACH

4.1 Outline

Existing approaches to the selection process from a large set of possible actions consists of reorganizing and navigating through the set, according to either the words of a sentence or the facts describing the state of the fictional worlds. The proposed alternative, *PastMaster@Storytelling* involves the use of recording the previously actions in temporal order, and using this temporal order as a key element for the organization of the set of possible actions.

PastMaster@Storytelling presents the list of all events that has been executed and displayed so far by the user and the system, from the beginning of the play to the present time. In most dramatic forms, the past events are not shown to the user. The user perceives only the current event, and is not able to access to past events. In a book on the contrary, past events are available, by reading previous pages. In Comics, a form of Drama, the past events are also available.

It is possible to include past events in computer–based storytelling systems. Most computer games for example, provide access to the saved games, even though they do not allow a full access to the past events.

Making past events accessible in a text form allows the user to click on pieces of texts to trigger a list of choices related to the selected text.

Indeed, in a narrative, possible actions are always related to past events. Therefore choosing the action through the past events is rather natural.

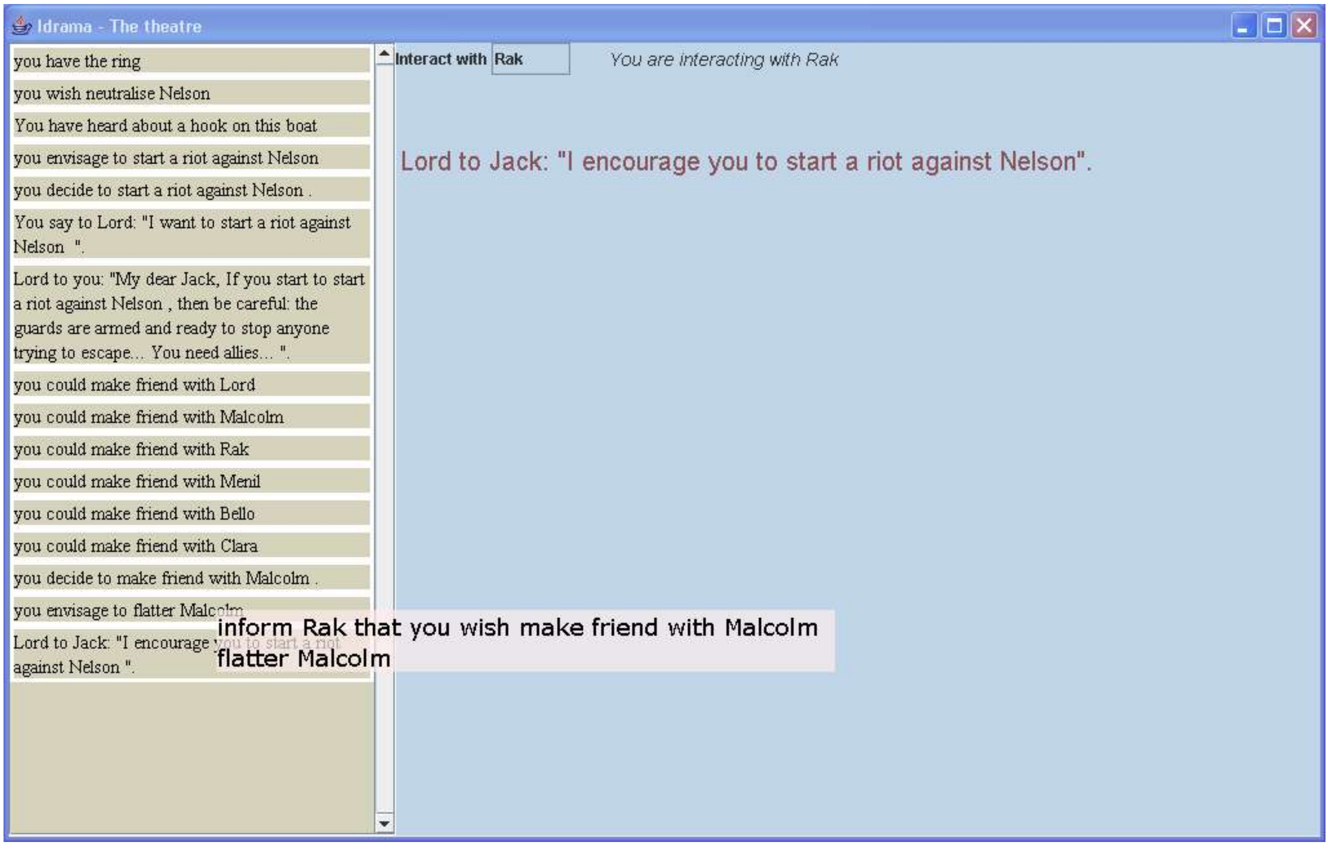
For example, suppose that the action "You inform Mary that you want to steal the key from Bob" has been executed previously. By selecting the text "you want to steal the key from Bob", all possible actions involving this text will be proposed, for example "inform *Ann* that you want to steal the key from Bob". A screenshot of the Interactive Drama engine is reproduced on Figure 3. The user is proposed two choices by clicking on a specific word in *PastMaster* (left panel in the interface).

PastMaster can be considered as a two level hierarchical menu, which is adaptive in two ways:

- First level adaptability – The list of events in *PastMaster* is always growing. This offers more and more selectable texts to the user.

- Second level adaptability – Following a given piece of text selected by the user, the list of proposed actions depends on the context. For example, if the possibility to steal something is not yet known by the user, then this task will not be proposed when the user selects "key" on *PastMaster*.

In addition, to limit the number of choices provided at the second level, the user chooses who s/he is interacting with (the addressee), using a simple choice list (see Figure 3). Note that in a future graphic implementation of the stage (typically in real time 3D), the selection of the addressee will occur naturally, through the user's navigation.



you have the ring

you wish neutralise Nelson

You have heard about a hook on this boat

you envisage to start a riot against Nelson

you decide to start a riot against Nelson .

You say to Lord: "I want to start a riot against Nelson ".

Lord to you: "My dear Jack, If you start to start a riot against Nelson , then be careful: the guards are armed and ready to stop anyone trying to escape... You need allies... ".

you could make friend with Lord

you could make friend with Malcolm

you could make friend with Rak

you could make friend with Menil

you could make friend with Bello

you could make friend with Clara

you decide to make friend with Malcolm .

you envisage to flatter Malcolm

Lord to Jack: "I encourage you to start a riot against Nelson ".

You are interacting with Rak

Lord to Jack: "I encourage you to start a riot against Nelson".

inform Rak that you wish make friend with Malcolm

flatter Malcolm

Figure 3. Screenshot of *PastMaster*. The user is talking to Rak about Malcolm.

The detail of the adaptation mechanisms are provided in Section 4.3.

4.2 Advantages and limitations

PastMaster@Storytelling presents several advantages regarding the Choice Problem mentioned above.

First, the interaction occurs on a module which is naturally integrated within the Interactive Drama. A play history, interactive or not, is a useful feature in Interactive Drama, especially if complex plots are involved. Many video games use similar features, through the concept of a diary, which automatically collects significant events in the game (e.g. *Baldur's Gate*, *Shenmue*). Thus, the use of *PastMaster* as a basis of interaction is transparent. It does not overwhelm the user by adding another layer in the interface.

Second, most of the actions in a narrative refer to what happened just before. For example, in an object-based interface, to answer the question "Do you have the key?", the user would need to find the key in the list of existing objects (his inventory if s/he owns the key) and then selects the appropriate answer. In *PastMaster*, the word "key" has just been pronounced, and is immediately available. Thus, the interface makes it easy to find most obvious actions. Furthermore, assume that the interface is properly designed, it would encourage the user to act according to the recent events, which improves the overall narrative experience (see the Thought Flow criterion for Interactive Drama in [21]).

Third, this approach promotes a "narrative way of thinking". Instead of reasoning in terms of goals and states, the user tends to reason in terms of events, i.e. in terms of transformation between states. For example, to ask another character for the key, the user would have to think "when did somebody mention the key?". In some cases, it might not be the most direct access. However, it would encourage the user to relate events from the past to future actions.

There are two limitations with *PastMaster* approach.

The first limitation is that the breakdown of actions is limited to two levels. While the sentence-based approach can deal with an arbitrary hierarchical complexity of sentences (at the cost of additional clicks), *PastMaster* approach might not solve the Choice Problem in case of very large sets of possible actions.

The second limitation is that the first level of the dynamic navigation system will be large. After a while, the user might find it hard to find a precise event in the past.

4.3 Related approaches

In the general field of HCI, interaction histories has been used and investigated for a long time as a support for the user. The various research studies can be classified according to the way they make use of the interaction history.

The first group of these approaches are the one that do not display the interaction history, although they do make use of it. In collaborative filtering or other usage-based personalization techniques for example, the interaction history enables a smart system to predict the user's preferences. A similar approach is used in context aware interfaces. The interaction history concept is exploited to determine the user's context of work to be able to provide the user with task-adapted services.

The second group demonstrates the interaction history to the user for supporting his/her activities. However, the user's interaction with the history is limited to navigation capabilities, and it neither triggers new actions, nor provides the system with data to trigger future actions. The Footprints project, for example, consist of attaching group usage information to the objects manipulated by the user in a browsing activity [20]. More recently in [13], a personal diary is provided to support the user. As mentioned above, the same idea of a diary is used in several computer games as a metaphor for a play history.

The third group consist of using the interaction history as a tool for the user to enter some data or commands. In [19], the interaction history is not only visualized and navigable, but also enables the user to inform the system regarding which past cases reflect his/her preferences more closely. In most systems of this kind, the interaction history is used to redo or undo a sequence of commands, with possible variations between the past sequence and the new one. For example, the interface agent described in [12] the user selects an item in the interaction history (called a segment). This triggers a new set of possible commands in the communication menu with the agent such as "stop a goal", "undo a goal" or "replay a goal with a different context". In [8], the interaction history is used to enhance explanations in a tutoring system. The learner can select a past explanation of a tutoring system and ask the system to compare it to the current explanation. The system then automatically generates a text comparing the two situations, in order to improve the learner's understanding.

Our approach belongs to this latter group, but it differs from existing systems on three points:

- The interface history is used as the “one and only” access point to the various commands in the system. Consequently, it is used to access many commands, not only a small set of specific commands (like undo/redo, for example).
- The interface is adaptive: the set of proposed action from a given item in the interaction history is changing in time, according to the context.
- Interaction occurs *within* each temporal event in the interaction history. This means that the contextual menus are not based on the whole temporal event, but on subparts of this temporal event, that is pieces of text.

4.4 Details of the contextual adaptation

The contextual adaptation consists of matching pieces of text in the history with a set of possible actions. Inside a past action in the history described as text, sub-parts must be delimited and linked to possible actions.

For example suppose that the history contains the action "Mary informs you that Bob wants to steal the key from Allan" and that the set of action contains "inform Helen that Bob wants to steal the key from Allan", "ask Helen if she has the key". Then the matching is as follows:

– Bob wants to steal the key from Allan -> inform Helen that Bob wants to steal the key from Allan

– steal the key from Allan -> inform Helen that Bob wants to steal the key from Allan

– key -> ask Helen if she has the key, inform Helen that Bob wants to steal the key from Allan

This means that if the user clicks on "Bob wants to steal the key from Allan", he/she is proposed "inform Helen that Bob wants to steal the key from Allan".

The matching is done automatically by exploiting the logical form of actions which serves as a basis for the generation of text description of actions. Indeed, an action is represented through a predicate form, for example:

Inform(Mary,John, want(Bob,steal,(key,Allan)))

Inside a given action, the elements which the user can select are:

– simple entities such as characters, objects, places, goals, tasks, obstacles (see [17] for a description of the models used in the Interactive Drama engine), except the user's character (to avoid too many choices in that case)

– facts, that is predicates describing the states of characters and objects, such as want, know, has-finished, etc.

The text generated from the logical forms of actions will include special markers so that the text corresponding to the two types of elements above can be selected by the user (see Section 4.6 for the detail of how the selection is physically performed).

The selectable areas are fixed during the game. This means that if there are no actions linked to the selected text, then a message such as "Nothing to do from this" is displayed.

Conversely, the list of possible actions is changing continuously, and the link between the selectable areas and the actions are recalculated at each turn. The following rule is used:

IF a possible action

(1) contains the element corresponding to a selectable area

(2) is addressed to the specified addressee

OR is a performance

OR is addressed to nobody and no addressee is selected

THEN the selectable area is linked to this action.

The second precondition consists in filtering the proposed actions according to the character the user is interacting with.

The interface is redundant, at two levels:

– the same element is clickable at various events in the history. The same set of proposed actions can be displayed for any of those;

– because elements are often embedded in other elements, the set of actions linked to the embedding element is also linked to the embedded element. In the example above, "key", "steal the key from Allan" and "Bob want to steal the key from Allan" are all three linked to "inform Helen that Bob wants to steal the key from Allan". In that case of embedded selectable areas, the larger the area, the more specific the set of proposed actions.

This latter type of redundancy could be avoided, but it would be disturbing for the user not to be able to select some actions via its elements, only because an embedding element allows this selection as well.

4.5 Specific actions in *PastMaster*

All actions calculated by the narrative engine are accessible via *PastMaster*. With the original narrative engine, this was not the case, and some adaptations were needed.

The first case is the beginning of the narrative: Obviously, there are no past cases of actions executed. To cope with this problem, *PastMaster* is initialized with a back-story, that involves items describing previous actions as well as characters, objects or locations independent of any actions. For example, one item of the back-story would be "You have heard about a dog in this house", which makes it possible to talk about this dog.

The second case is the creation of new tasks and goals during the narrative (tasks are concrete activities to reach goals, see [17] for details). In certain conditions, the user has the possibility to decide to reach new goals and to perform new tasks. These actions are not easily related to previous actions in the narrative. Thus, each time some new goals or tasks appear in the story, some implicit internal actions, or thoughts are added. Two kinds of such actions are used: "envisage a goal" and "envisage a task". Those actions are automatically added to the history when new goals or tasks are made available to the user. It is then possible to the user, by selecting sub parts in those actions, to perform actions related to those goals and tasks.

4.6 Visual design of the interactive play history

Interacting with the play history requires being able to select parts of texts, as specified in the previous section. The ergonomical difficulty lies in the fact that those parts of texts are possibly embedded in each other.

For example, in the previous example, putting in brackets the clickable zones leads to the following result:

> [Mary] informs you that [[Bob] wants to [steal the [key] from [Allan]]]

To make it possible to the user to select the proper zones the following design principles have been taken:

- When a text is rolled over by the cursor the corresponding area gets highlighted (both underlined and italicized)
- When the cursor is in a zone belonging to more than one logical element (embedded elements), only the smallest one is highlighted.
- A click on the highlighted zone triggers the associated list of possible actions (see previous section).

The list of possible actions is displayed in an overlapping semi-transparent window (see Figure 3).

5. PRELIMINARY EVALUATION

5.1 Goal of the evaluation

The type of interface that is proposed in this paper is a novel approach, both in video games and HCI in general. Two general issues must be investigated before refining the interface and conducting large scale evaluations.

First, the global usability of the interface has to be assessed. It might be the case that users completely misunderstand *PastMaster* and are not able to manage to choose actions meaningfully.

Second, the cognitive processes involved in the use of the interface must be investigated. Following versions of the interface should be designed in order to support the cognitive processes that are best suited to Interactive Drama.

A set of pilot experiments have been conducted in order to resolve these issues. Regarding the first issue (usability), our main golas is to answer the following question: Are the users blocked at some point, because they do not know what to do in the story?

Regarding the second issue (cognitive processes), our main goal is to find out whether users would have something to execute in mind prior to the interaction and they use the interface to execute this action or whether they explore the interface in order to see possibilities. The former case is preferred, because it means that the user is more engaged in the story and less in the interface.

5.2 Protocol

The pilot experiments comprise three phases:

(1) Introduction, explanation of the principle of Interactive Drama and explanation of the user interface (10 minutes)

(2) Interaction with the system, during 10 minutes. The story's setting can be summarized as follows: "During the mid 17th century, a Galleon is navigating from the Caribbeans to the Old Continent. On board are four prisoners. One of them, acted by the player, intends to start a riot". The user is allowed to ask questions to the investigator, seated next to him/her.

(3) Debriefing and filling out a questionnaire (10 minutes)

Four subjects were recruited, two males and two females. None of them was a "gamer", but all were computer literate.

The investigator was able to write down qualitative observations about the interactive session. A video capture of the session was performed for further analysis. The questionnaire filled out contains 8 personal questions and 12 questions regarding the interface itself.

5.3 Results

5.3.1 Ease of use

All subjects managed to interact with Interactive Drama Engine via *PastMaster* for 10 minutes. The rounded average of user actions (or turns) was 18.

From the qualitative observation of the interaction, no subject has been lost in navigation by the complexity of the interface. In the questionnaire, when asked whether the interface was easy to use, two strongly agreed, and two agreed (within five choices from strongly agree to strongly disagree). From both observation and analysis of the questionnaire, it was found that subjects understood quickly, in less than five turns.

However, when asked more precisely if it was easy to choose an action, results were less positive. Only one subject found it very easy. One subject found it hard.

Another question in the questionnaire revealed that two subjects found it often hard to select a past action in *PastMaster*. They both suggested ways to access those actions more easily (by grouping them in threads, for example)

5.3.2 Cognitive processes

When asked if they thought of a specific action in mind to execute prior to interacting with the interface, two subjects answered “sometimes”, two ”seldom”. This means that the subjects were most of the time exploring the interface to find actions to play, rather than the other way around.

One of the subjects was spontaneously talking aloud, providing additional information. Some talks confirmed that the subject was sometimes thinking of an action beforehand. In other cases, the subject has an idea of action in mind but this action did not fit exactly with the framework used by the narrative engine.

Analysis of the free comment about their experience revealed that some subjects relied on the engine to provide some interesting happenings and were disappointed. Typically, one subject would repeat the same action, which led to an obstacle every time, hoping that the result would be different next time; the subject then complained to be in a loop. Similarly another subject did repeat the same actions several times and then explained that s/he "was expecting something different to happen".

5.4 Discussion

Before discussing the results themselves it should be noted that even with four subjects, high variations in user behavior are observed. This may be explained by the original nature of the task, for which no usual and more or less standardized procedure was applicable.

These preliminary tests have confirmed the validity of the approach, namely the idea of accessing actions via an adaptive menu based on the past events in the narrative.

It is difficult to access the relevant past actions in the history. The fact that in the current interface, those actions are displayed in a flat manner leaves plenty of room for improvement. Several ways of structuring the list of actions can be suggested: temporal clustering (by scenes, days), relevance-based marking, content-based access, etc.

Infrequently users were thinking an action prior to interaction. This is not that surprising, because the subjects were not told about the range of actions handled by the narrative engine. One subject for example commented that s/he expected his/her choice to be limited so s/he preferred to look at the menu first. Progressive explanations of the range of possible actions may form a way of improvement (for example, it is possible to inform any other character about what one knows).

These pilot experiments also give us guidelines for designing a large scale evaluation of the interface. In particular, a thinking-aloud protocol should suit this investigation, because in case of narrative, it is hard to interpret the user's mental model just by observing the behavior.

6. APPLICATION TO OTHER DOMAINS

6.1 Narrative Intelligence

Although interaction histories are in use from decades, through the notions of traces and logs, graphical user interfaces make a limited use of such facilities. According to V. Kaptelinin, "both research and practical applications of interaction histories are still in their infancy" [6]. In best cases, interaction histories have a limited and optional role in the interaction.

Narrative Intelligence however is a new tendency to structure the user experience with computers in a temporal and narrative manner [9]. It is based on the assumption that narrative is a fundamental way for human beings to understand and organize their experience [1].

In this theoretical view *PastMaster*, for it allows the user to explicitly base his/her future action on past actions, provides a practical way to support the narrative construction of interactivity, to paraphrase J. Bruner famous essay [1]. In other words, any software whose activity is complex and spread over time would benefit from *PastMaster*, in terms of cognitive organization of the user experience. The approach should help diminishing the cognitive overload syndrome which occurs when many tasks are performed in parallel, as it often the case today.

PastMaster is particularly suited when various and complex commands are available which can be contextually initiated from past actions. We shortly review two of these situations which occur in everyday computer activity: information search and personal e-mail management.

6.2 Information Search

Many search engines, including *Google,* give the possibility to store the history of search queries. Typically those queries can be viewed, deleted, edited and restarted. This set of commands is fixed, independent of the current context.

Using our approach, intelligent support to the web searching could be provided by enabling contextual commands from subparts of past queries.

Suppose that a history is dynamically constructed, which comprises both queries and documents viewed. Three effective usages of this history are suggested, using *PastMaster*:

- Just after having queried the search engine, the user clicks on other terms in past queries and is proposed to search a new query, which is the combination of the current query and the selected term (AND combination).
- The user just displayed the query list of results which happens to be too large and not focused. By clicking on a previously viewed document, s/he is propose to refined the search by adding the past document in the query (document based query).
- The user is viewing a document. If s/he clicks on a search term in the past queries, s/he is proposed to highlight the term in the current document.

Note that in this application, the events in history are much simpler than in the Interactive Drama case.

6.3 Personal e-mail management

E-mail softwares store e-mails in a chronological order, which constitutes a history. However, it is not an interaction history. An interaction history would contains events such as:

> "you read an e-mail received at 1:07 from Mary entitled 'About last meeting', with Helen in CC"
> "You answer to Mary's 1:07 e-mail ('About last meeting'), with John in CC"
> "You move Mary's 1:07 e-mail ('About last meeting') to the 'Meetings' folder

Suppose that the user is then writing to Helen, then the following commands could be triggered via the interaction history above:

1. By clicking on the second event, the user is suggested:
   - "tell Helen that you have answered to the e-mail from Mary (with the answer)"
   - "advice Helen to read this e-mail carefully".

If the user clicks on one of the suggestions, a paragraph is automatically inserted in the current e-mail.

2. By clicking on "John" only inside the same event, the user is suggested:

- "Send an e-mail to John"
- "Add John to the addressee list"
- "Add John to the CC list".

Those examples illustrate how complex commands can be easily built via the interaction history. Tailored to a specific need of a community of e-mail users, they might greatly improve the group collaborative work and communication.

7. CONCLUSION

In the context of Interactive Drama, *PastMaster*, a new Graphical User Interface has been proposed based on the interaction history. *PastMaster* aims at solving a major issue of Interactive Drama, the large number of choices proposed to the user. *PastMaster* is a context-adaptive hierarchical menu which enables to propose a large number of actions with a minimal overwhelming interface.

Preliminary tests tend to show that the principle of this kind of interface is accepted by the user. However, in its current stage, *PastMaster* must be improved to make easier the retrieval of past actions in the history.

The narrative intelligence that is embedded in the narrative engine and its interface discussed in this paper is complementary to the intelligence classically used is advanced interfaces. The latter aims at understanding user's action, while the former aims at shaping user's action in a meaningful way. There is a future in combining the two approaches. It would consist for example in understanding user's speech to find the past action. The speech recognition would become feasible because it supports the user in relating to past actions, that is in setting his/her context, rather than attempting to solve the problem of Human understanding.

8.ACKNOWLEDGEMENT

This research is supported by an *Australian Research Council* Linkage International Fellowship Grant titled "An Interactive Drama Engine in Virtual Reality" (#LX560117).

9.REFERENCES

[1] Bruner, J. S. The narrative construction of reality. Critical Inquiry, 18 (1991), 1-21.

[2] Cavazza, M., O. Martin, F. Charles, S. J. Mead, X. Marichal. User Acting in Mixed Reality Interactive Storytelling. In *Proc. of the 2nd Int. Confe. on Virtual Storytelling (ICVS 2003)* (Toulouse, France), LNCS 2897. Springer Verlag, 2003, 189-197.

[3] Crawford, C: Assumptions underlying the Erasmatron interactive storytelling engine. In *Papers from the AAAI Fall Symposium on Narrative Intelligence*, Technical Report FS-99-01. AAAI Press, Menlo Park, CA, 1999, 112-114.

[4] Crawford, C. Deikto. http://www.erasmatazz.com/Erasmatron4/Erasmatron4.html.

[5] Genette, G. *Figures III*. Seuil, Paris, 1972.

[6] Kaptelinin, V. Managing project contexts: Interaction history as a ressource. In *Proc. of the Workshop on Exploiting Context Histories in Smart Environments (ECHISE'05)*, (May 11, 2005, Munich, Germany). 5-10.

[7] Laurel, B. *Computers as Theatre*. Addison-Wesley, Reading, MA, 1993.

[8] Lemaire, B., and Moore, J. An improved interface for tutorial dialogues: Browsing a visual dialogue history. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'94)* (Boston, April 24-28 1994). ACM Press, 16-22.

[9] Mateas, M. and Sengers,P. Narrative Intelligence. In *Papers from the AAAI Fall Symposium on Narrative Intelligence*, Technical Report FS-99-01. AAAI Press, Menlo Park, 1999.

[10] Mateas, M., and Stern, A. Natural Language Understanding in Façade: Surface-text Processing In *Proceedings TIDSE'04*, LNCS 3105, Springer Verlag, 2004.

[11] Murray J. *Hamlet on the Holodeck*. The future of narrative in the cyberspace. Free Press, New York, NY, 1997.

[12] Rich, C., and Sidner, C. Segmented interaction history in a collaborative interface agent. In *Proceedings of the 3rd Int. Conf. on Intelligent User Interface (IUI'97)* (Jan. 6-9, 1997, Orlando, FL). ACM Press, 23-30.

[13] Schneider, M., Bauer, M., and Kröner, A. Building a Personal Memory for Situated User Support. In *Proc. of the Workshop on Exploiting Context Histories in Smart Environments (ECHISE'05)* (May 11, 2005, Munich). 43-48.

[14] Schwitter, R., Ljungberg, A., Hood, D. ECOLE - A Look-ahead Editor for a Controlled Language. In *Proceedings of EAMT-CLAW03* (May 15-17, 2003, Dublin), 141-150.

[15] Stern A., and Mateas, M. Integrating Plot, Character and Natural Language Processing in the Interactive Drama Façade. In Göbel et al. (eds) *Proc. TIDSE'03*. Frauenhofer IRB Verlag, 2003.

[16] Szilas, N. Interactive Drama on Computer: Beyond Linear Narrative. In *Proc. AAAI Fall Symposium on Narrative Intelligence* (North Falmouth MA, November 1999), AAAI Press, 150-156.

[17] Szilas, N. IDtension: a narrative engine for Interactive Drama. In *Proceedings TIDSE'03*, Göbel et al. eds., Frauenhofer IRB Verlag, 2003.

[18] Szilas, N. Stepping into the Interactive Drama. In *Proceedings TIDSE'04, Lecture Note in Computer Science 3105,* Springer Verlag, 2004, 101-112.

[19] Terveen, L., McMackin, J., Amento, B., Hill, W. Specifying Preferences Based on User History. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'02)* (Minneapolis, MN, April 20-25 2002). ACM Press, 315-322.

[20] Wexelblat, A., and Maes, P. Footprints: History-Rich Tools for Information Foraging. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)* (May 15-20, 1999, Pittsburgh, PA). ACM Press, 270–277.

[21] Weyhrauch, P. *Guiding Interactive Drama*. Ph.D. Dissertation, Tech report CMUCS-97-109, Carnegie Mellon University, 1997.

[22] WYSIWYM project. http://www.itri.brighton.ac.uk/projects/WYSIWYM/. Accessed 17/09/2005.

[23] Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R.J. Saretto, C. J. An architecture for integrating plan-based behavior generation with interactive game environments, *Journal of Game Development*, 1, 1 (2004).