Adaptive Real Time Comment Generation for Sail Racing Scenarios

Andrea Esuli[1], Antonio Cisternino[1], Giuliano Pacini[2], and Maria Simi[1]

[1]Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy
{esuli, cisterni, simi}@di.unipi.it
[2]Accademia Navale di Livorno, Livorno, Italy
pacini@di.unipi.it

Abstract. We describe a comment generation subsystem developed as part of an expert authoring system for sail racing scenarios. The result of the expert system analysis is sent to a multimedia presentation subsystem composed of a 3D player and a speaker agent. Comments output by the text-to-speech component must be carefully generated so that they are concise, relevant, non repetitive and timely since they must be synchronized with the 3D animation. The comment generation process is adaptive with respect to the time available and to the user profile and preferences.

1 Introduction

As part of our work in developing intelligent tutoring systems for sail racing, we designed and implemented a system for the multimedia presentation of scenarios involving dynamic objects in general [1]. The architecture of the system, (see Fig. 1) was inspired by the simulated soccer system within the RoboCup initiative [2].
Each module is implemented as a program that interacts with other modules using a TCP connection and a text protocol.

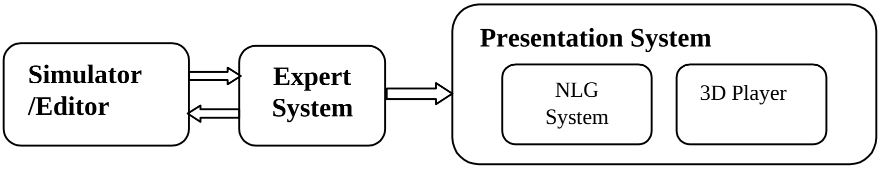

Fig. 1. The architecture of the expert authoring system

The authoring of sail scenarios is performed by means of a visual editor, working in close cooperation with a simulator of boats movements. The editor offers a number of high-level functionalities specific of the sail racing domain (such as *combining trajectories*); the simulator ensures that the dynamic scenarios comply with physical constraints. The simulation *history*, output of the *Simulation/Editor* module, consists of a stream of instantaneous snapshots describing the boats positions (bi-dimensional) and status (direction, rudder and sails).

P. De Bra, P. Brusilovsky, and R. Conejo (Eds.): AH 2002, LNCS 2347, pp 392–396.
© Springer-Verlag Berlin Heidelberg 2002

The role of the *Expert System* module is to analyse and annotate with comments histories received by the editor, using knowledge about the racing rules. In some cases, in order to verify some conditions, the expert system needs to ask the simulator how a given hypothetical situation will possibly evolve. The expert system analysis results in a structured representation of all the facts worth commenting about, marked with their importance. The expert however is not in the position to do a detailed planning of the comment since it does not know about the rendering media, time constraints or user preferences.

The *presentation system* (see Fig. 2) consists in a 3D player that is responsible for showing the simulation and uttering the comments generated by the comment generation module. A Web browser with a VRML plug-in is used to render the virtual scene, under the control of a Java applet through the *External Authoring Interface*. The main control functions are written in Jscript. The Microsoft Agent technology is used to implement the animated character [3].

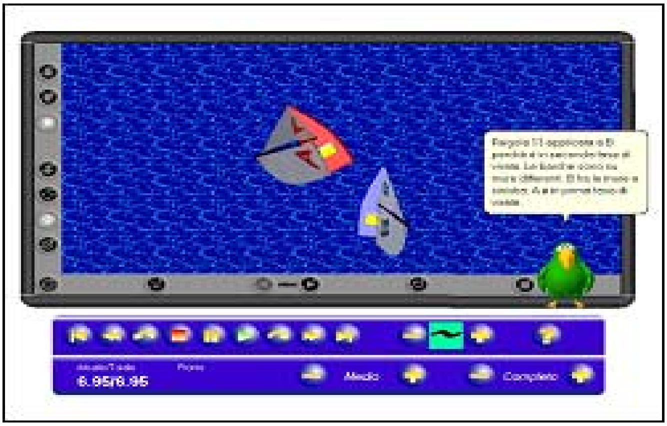

Fig. 2. Player interface

The comment generation sub-system needs to perform a complex natural language generation task in close cooperation with the visual rendering and the user. It is responsible for generating the most informative messages that can be pronounced in the available time and to pass control to the Microsoft Agent, who will output them using a text-to-speech component.

The rest of the paper will give more details about the strategy of adaptive comment generation implemented.

2 The Comment Generation Subsystem

The language generation task consists in producing meaningful natural language comments from *comment structures* provided by the expert system for each snapshot of the scenario. This process has to take into account temporal constraints imposed by the dynamics of the scene and other parameters such as the level of expertise of the users and their preferences, expressed by suitable controls in the interface.

As traditional in NL generation [4,5] this task is performed in two phases. First, a contents selection phase is performed in order to discard repetitions, select and organize the concepts to be communicated. In the second phase, concepts are rendered as natural language text [6]. This two phases however are not strictly sequential; in fact contents selection needs to be performed again, after text generation, when the time available does not allow completion of the generated comment. We regard this interplay of the contents selection and generation phases as one of the main characteristics of our approach.

2.1 Contents Selection

A comment is represented by a structure whose basic elements are *primitive facts*, represented as sentences composed of a subject (`s`), a predicate (`p`) and an object (`o`). Two parameters are associated to each primitive fact:

- the *importance*: a value rating the relevance of the fact in the present context;
- the *difficulty level*: specifying the language complexity class for which the fact is appropriate, ranging from a highly specialized language to an ordinary language.

Here is an example of primitive fact, represented in Jscript syntax:

```
{s:″B″, p:″is″, o:″beyond herd to wind″, i:400, d:[0,1]}
```

where `i` is the importance value and `d` assigns difficulty levels to the fact.

Each comment structure is composed of a number of primitive facts related by predefined relations, namely:

- *Facts-Additional facts* (FA): additional facts are complements to the main facts
- *Facts-Explanations* (FS): explanation facts further describe main facts.
- *Cause-Effect* (CE): cause-effect relation between the two groups of facts.

The expert system produces a comment for each fired rule, thus the same primitive facts may appear in more than one comment. A pre-processing step takes care of collapsing such duplicate facts into one instance. As a consequence, facts belonging to different rules may become related.

The main task in contents selection is the selection based on the *richness level*, as selected by the user through the interface. This is based on a threshold value (similar to [7]), computed as the weighted sum of a static factor and a dynamic one. Primitive facts that have an importance level above the threshold pass the selection.

The *static factor* is a percentage, which depends on the richness of the comment, of the maximum importance value of the facts in the current snapshot. The *dynamic factor* is a percentage of the threshold value of the previous snapshot and it is used for allowing the completion of important comments at the expense of successive less important ones (Fig. 3). This aspect is strongly related to the extension strategy described next.

2.2 Comment Generation

In principle selected comments should be all presented but often this is impossible due to time constraints. The generation phase consists in a cycle where comments are progressively discarded until temporal constraints are satisfied. At each iteration two main steps are performed for comment generation:

- *Linearization*: the selected facts are ordered according to their relations in order to construct a plan of the comment sentence (*text design*).
- *Realization*: the natural language sentence is produced according to the previous linear structure (*text realization*).

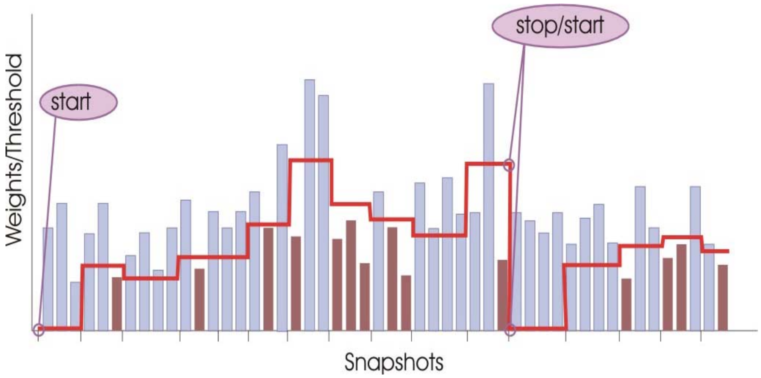

Fig. 3. Evolution of the threshold.Discarded comments are darker.

At the end of each iteration if the generated text comment can be output during the animation time (according to an experimental estimation) the cycle is terminated and the sentence is sent to the text-to-speech synthesizer of the Microsoft Agent. Otherwise an attempt is made to extend the comment to successive snapshots.

This *extension process* activates the selection phase on the next snapshot; if no comment remains after selection, this snapshot is marked as *silent* and the time available to pronounce the comment is extended with the animation time of this snapshot. If the time is still not sufficient, the function is called recursively on successive snapshots until the end of the planning window is reached.

If the necessary time cannot be found by this strategy, two alternatives remain:

- In case of a reproduction modality that we call *guided by the comment*, the commentator slows down the animation by increasing by a small percentage the animation time and repeats the extension cycle. No comment is discarded.
- If the reproduction modality is *fixed*, the commentator is forced to discard the least important comment in the current snapshot and the generation cycle is repeated.

The problem of real time comments has been tackled by RoCCo (RoboCup Commentator) in the context of simulated soccer [8]. Our approach to comment generation however is not based on text templates but on concept structures more similar to the ones used for example in the ILEX system [9].

References

1. Cisternino A., Del Cioppo E., Marotta P., Pacini G., Silenzi A., Simi M.: An expert authoring tool for dynamic scenarios, *Proc. of ED-MEDIA*, (2001)
2. Kitano, H., Asada, M., Osawa, E., Noda, I., Kuniyoshi, Y., Matsubara, H.: RoboCup: A Challenge Problem for AI, *AI Magazine*, Vol. 18, No. 1, (1997)

3. Microsoft Agent Software Development Kit, Microsoft Press, Redmond, (1999)
4. Cole, A., et al: Survey of the State of the Art in Human Language Tecnology, Kluwer, Dordrecht, (1996)
5. Wahlster: Intelligent Multimedia Interface Agents, Tutorial MICAI 2000, (1999)
6. Feiner S.K., McKeown K.R.: Generating Coordinated Multimedia Explana-tions, *Proceedings of the 6th IEEE Conference on A.I. Applications*, (1990)
7. Fraczak L., Lapalme G., Zock M.: Automatic generation of subway directions: salience gradation as a factor for determining message and form, *Proceedings.of the Ninth International Workshop on Language Generation*, Canada, (1998)
8. Voelz D., André E., Herzog G., Rist, T.: RoCCo: A RoboCup Soccer Commentator System, *Lecture Notes in Computer Science*, Vol 1604, (1999)
9. Oberlander J., O'Donnell M., Knott A., Mellish C.: Conversation in the museum: experiments in dynamic hypermedia with the intelligent labelling explorer, *New Review of Hypermedia and Multimedia*, Vol. 4, (1998) 11-32