

# Automated Commentaries for Simulated Soccer

## Priority Evaluation

Audience	All
Author	Justin Hogg
Scope	
Date Created	05/03/2007
Version Number	0.1
Version History	0.1 – Original Document (JH)
Reviewed	Yes
Last saved by Justin Hogg, 18/03/2007	

# **Sign-off sheet**

**Date: 18/03/2007**

**Document Author Signature: JH\***

**Document Author Name: Justin Hogg**

**Quality Assurance Signature: AM\***

**Quality Assurance Name: Adrien Martel**

**Project Manager Signature: AM\***

**Project Manager Name: Ahsan Mussa**

\* By signing this document you approve that the entire contents of the deliverable has been reviewed and is in line with the objectives of the project.

## **Priority Evaluation by Justin Hogg**

### **Technical Details**

Text based comments are marked up with variables by the Commentary Producer class, and passed to the Comment Scheduler class where they are added to the proposition pool – a hashtable of all possible comments to be spoken based on game events that have occurred. Every three seconds, the separately executing thread of the Comment Scheduler selects the comment from the pool with the highest importance rating and passes it to the voice classes for audio output; it also provides a parameter that defines the voice excitement level for that comment. The pool is then emptied, and in three seconds time, the process repeats. The CS can also include a parameter to indicate which of two possible voices will output the speech – there is one for statistical comments and one for game event comments.

Two threads need to access the proposition pool – the thread running the Commentary Producer class to add text comments to the pool, and the separate Comment Scheduler thread to select comments for audio output and delete the pool contents. Due to this fact, access to the hashtable that implements the proposition pool is synchronised to eliminate race hazards associated with shared resources in a multi-threaded environment.

The thread that runs the Comment Scheduler polls continuously to determine when the next comment should be output. It also tests when lower priority comments (the most frequently occurring) were added to the proposition pool, and only outputs them to audio if they were added within the last 100ms – this ensures lower priority audio comments are synchronized with the real time game events taking place on the soccer monitor. Higher priority comments are output regardless of their age in the pool, as a slight lag is preferable to their not being mentioned at all. This lag has been minimised through a number of different iterations of the Comment Scheduler class, along with a number of further optimisations for the most effective, real time audio commentary. For more details, see document: CommentScheduler0.1.

### **Problems/Outcomes**

The final implementation offers a detailed and informative audio commentary that is synchronised with the events taking place on the soccer monitor. All comments relating to high importance events i.e. goals and shots, are output as audio. Less important, high frequency events i.e. passes, loss of ball possession or dribbling with the ball, are not always output as audio. This is due to the fact that the time taken for an audio comment to be spoken exceeds the time duration of the actual event on the soccer monitor. If every comment is output as audio, the commentary will fall behind the game events taking place on the soccer monitor.

The design of each comment and the prioritisation is optimised to give a balance between a richer, more detailed commentary and still providing

commentary on the majority of game events. Two voices are used by the system to give the impression of two commentators.

Earlier implementations revealed additional requirements for the scheduling of comments. Without a test to determine when a comment was added to the proposition pool, the highest priority comment was always output. This is desirable in the case of a goal, or high importance event, even if the comment represents an event that occurred in the past. It is not desirable in the case of lower important events, to output an audio comment representing the event if it is not the current game event taking place on the field, as this compromises the real time delivery of the commentary. By assigning the majority of lower importance game event comments an importance rating of twelve (see document: Comment Priority Levels 0.2), older comments in the proposition pool with the same importance rating are overwritten and thus lower importance comments in the pool are kept as fresh as possible in relation to events taking place on the soccer monitor. Although this defines less of an importance hierarchy associated with comments (see document: Comment Priority Levels 0.1), it significantly improves the real time delivery of the audio commentary and this is the prime concern of the system.

Efficiency is also improved by not adding comments to the pool if they have a lower importance than the comment in the proposition pool with the highest importance rating. As access to the pool is synchronised between threads, the thread that does not have access to the pool is blocked i.e. must wait. By minimising the time a thread may have control of the pool, the waiting time for other threads is minimized, and the speed of access, and thus the system, is increased. This principle also improved when the Commentary Producer class blocked repeated method calls to the Comment Scheduler, relating to the same instance of a game event (see document: CommentaryProducer0.1) as it reduced the number of calls (and requests for a synchronized lock on the proposition pool) made by the Commentary Producer to the Comment Scheduler.

The final version increases CPU usage to around 70%, although it offers a noticeable improvement over the previous versions, in terms of real time synchronization with the soccer monitor.

These are the main outcomes and issues, although the document CommentScheduler0.1 gives further details.

### **Future developments**

Comments have varying lengths of the time taken for the whole comment to be output as audio. The system waits for three seconds (the length of the longest comment) from the start of an audio comment, before outputting the next comment.

This could be improved in future versions by providing the length in milliseconds, of any comment output as audio. The Comment Scheduler could then output the next comment as soon as the current one had finished. More

comments could be spoken, but the commentary would be almost continuous. The current implementation produces varying periods of silence between each audio comment, which increases the realism of the commentary. The commentary system is different from most scheduling optimisation problems as having the system process as many tasks (comments) as possible is not necessarily desirable as it would produce a continuous audio output and comments would blur together.

Processing as many tasks as possible is usually the goal of scheduling, such as making optimal use of CPU usage for processing the task queue. Here any idle time is undesirable and wastes clock cycles.

Also, as a further development, the Comment Scheduler could have an awareness of the ball position on the field, and proximity to players. This could be used to determine if a comment in the proposition pool represents the current game state.

Such functionality would also be very useful in deciding when to output statistical comments. If the ball was being passed around the central area of the football pitch, statistical comments could be output that offer information on player performance, attack patterns, previous game trends, and more.