Character Design for Soccer Commentary

Kim Binsted[1] and Sean Luke[2]

[1] Sony Computer Science Lab
3-14-13 Higashigotanda
Shinagawa-ku, Tokyo 141 Japan
kimb@csl.sony.co.jp
http://www.csl.sony.co.jp

[2] Department of Computer Science
A. V. Williams Building
University of Maryland
College Park, MD 20742 USA
seanl@cs.umd.edu
http://www.cs.umd.edu/users/seanl

Abstract. In this paper we present early work on an animated talking head commentary system called **Byrne**. The goal of this project is to develop a system which can take the output from the RoboCup soccer simulator, and generate appropriate affective speech and facial expressions, based on the character's personality, emotional state, and the state of play. Here we describe a system which takes pre-analysed simulator output as input, and which generates text marked-up for use by a speech generator and a face animation system. We make heavy use of inter-system standards, so that future versions of Byrne will be able to take advantage of advances in the technologies that it incorporates.

1 Introduction

Many natural systems have behaviour complex enough that people will tend to ascribe personalities to them, and use those personalities as flawed but powerfully predictive tools. For example, we might summarize a dog's behavioral tendencies as "eager to please" or "yappy and spoiled", and use this assigned personality to predict its future behaviour.

Designed characters — such as characters in films or novels — exploit this tendency, expressing their personality so as to manipulate the observer's expectations to the designer's ends. For example, the villain in a novel might sneer and speak in a menacing voice, cueing us to expect villainous behaviour. This expectation might be reinforced by more static characteristics, such as a strong Transylvanian accent or a cheekbone scar. Consistency between expression and action, and also between modalities of expression, contributes to a character's believability. Believability, in turn, contributes to the expected predictive value of the character's perceived personality.

M. Asada and H. Kitano (Eds.): RoboCup-98, LNAI 1604, pp. 22-33, 1999.

We are interested in the relationship between consistency and believability, and between expression and perceived personality. To explore these issues, we are developing a talking head system which can generate entertaining, believable commentary on RoboCup simulator league games [11], complete with facial expressions and affective speech, in (close to) real time.

A goal of this research is to develop an architecture which allows artists (whose technological skills may vary) to design expressive, believable characters (in the first instance, talking heads). The emphasis is on the expression, not the content — we assume the pre-linguistic raw content is generated by another system, specialized to the application. In the context of a particular character, our system:

- generates appropriate natural language text to express that content,
- transforms that text into natural affective speech, and
- controls a face animation, so that appropriate lip movements and facial expressions are generated.

In this early stage of the project, we are mostly interested in the speech and facial animation components of the system.

The key issue here is *appropriateness*. How do we ensure that the behaviour of the system is appropriate for the designed character in the given situation? For example, when reporting a scored goal, the language and facial expressions used might depend strongly on which team the character supports. Moreover, how do we ensure that the designed character is appropriate for the application? A character which might be perfect for soccer commentary might not be right for, say, a medical advisory system.

This begs the question: is a talking head appropriate for soccer commentary at all? After all, the heads of human sports commentators are rarely seen on screen during the main action of the game. In fact, our attraction to this domain is due more to its usefulness for our research (please see Section 3.1) than because soccer 'needs' talking heads in any way. Nonetheless, we do believe that an expressive entertaining talking head commentary would add to the fun of watching (or in the case of of a video game, playing) soccer. This has yet to be seen, of course.

2 Related work

2.1 Believable characters

Recently there has been a great deal of interest in the design and implementation of characters with personality. For example, the Virtual Theatre Project at Stanford [8] is working on a number of animated virtual actors for directed improvisation, basing their efforts on Keith Johnstone's theories of improvisational theatre [9]. They make use of character animations developed as part of the IMPROV project [6], which can take fairly high-level movement directions and carry them out in a natural, expressive manner. Related work on agent action

selection and animation has been done as part of the ALIVE [3] and OZ projects [12] [16].

Although these projects have similar goals and assumptions to ours, our approach differs from theirs on several points. First, our focus on talking heads (rather than fully embodied agents in virtual environments) leads to a stronger emphasis on language-related behaviours. Also, we do not attempt to have the personality of the character control content selection, or any other action of the agent, for that matter. Although this sharp distinction between content and expression might negatively affect the consistency of the character, a clear separation between content and expression allows the character to be portable across content generators. For example, you could have essentially the same character (an aggressive older Scottish man, for example) commentate your soccer games and read your maps. In the first case, the content generating application is the RoboCup soccer simulator, and in the second case it is an in-car navigation system — but the character remains the same.

We also do not make a great effort to make the emotion component of our system cognitively plausible. Designed characters, such as characters in novels or films, are generally both simpler and more exaggerated than natural personalities, such as those we perceive in each other. The goal is not to develop a psychologically realistic personality, but to generate a consistent and entertaining character for some application. For this reason, what psychology there is in Byrne is more folk psychology than modern cognitive science.

2.2 Game analysis and commentary

There are at least two existing systems which generate analysis and commentary for the RoboCup simulation league: MIKE [20] and Rocco [1].

Rocco is a system for analysing simulation league games and generating multimedia presentations of games. Its output is a combination of spoken natural language utterances and a 3-D visualization of the game itself. The generated language has appropriate verbosity, floridity, specificity, formality and bias for the game context. It uses a text-to-speech system to synthesize the spoken utterances.

MIKE is a system developed at ETL which, given raw simulation data as input, analyses the state and events of the game, chooses relevant comments to make, and generates natural language commentary in real time and in a range of languages. It also uses a text-to-speech synthesizer to generate spoken output.

Our approach differs from the above in that we do not, at present, do any game analysis within the Byrne system itself; instead, we assume pre-analysed game information as input. This is because our emphasis is on expression, rather than utterance content. To our knowledge, neither MIKE nor Rocco attempt to generate affective speech, and neither use face animation.

3 Enabling technologies

In this section we outline some of the existing technologies that Byrne uses, and discuss some issues related to inter-system standards.

3.1 The RoboCup simulation league

The simulation league of RoboCup [11] [14] features teams of autonomous players in a simulated environment. This is an interesting domain for several reasons. There is no direct interaction with a user, which simplifies things a great deal — no natural language understanding is necessary, for example. More importantly, having direct access to the state of the simulation simplifies perception. If our system had to commentate a real (non-simulated) game, vision and event-recognition issues arise, which we'd rather avoid.

However, there are some problems with using the output of the RoboCup simulator directly:

- The output of the simulator is at a much lower level of description than that typically used by a football commentator.
- The simulator has no sense of which actions and states of play are relevant or interesting enough to be worthy of comment.

For this reason, some kind of game-analysis system is a necessary intermediary between the soccer simulator and Byrne.

3.2 Mark-up languages

SGML-based [7] mark-up languages are a useful tool for for controlling the presentation of information at a system-independent level. Here we make use of three different mark-up languages: one to indicate the linguistic structure of the text, one to determine how the text is to be spoken, and one to control the facial animation.

Global Document Annotation Global Document Annotation (GDA) is an SGML-based standard for indicating part of speech (POS), syntactic, semantic and pragmatic structure in text [13]. Although our main interest in this work is emotional expression, this must be underlaid by linguistically appropriate intonation and facial gestures. Moreover, the quality of speech synthesis is generally improved by the inclusion of simple phrase structure and part of speech information, which can be encoded in GDA.

Sable SABLE [17] is a SGML-based text-to-speech mark-up system developed by an international consortium of speech researchers. It is based on several earlier attempts to develop a standard, namely SSML [21], STML [18] and JSML [10]. The goal is to have a system- and theory-independent standard for text mark-up,

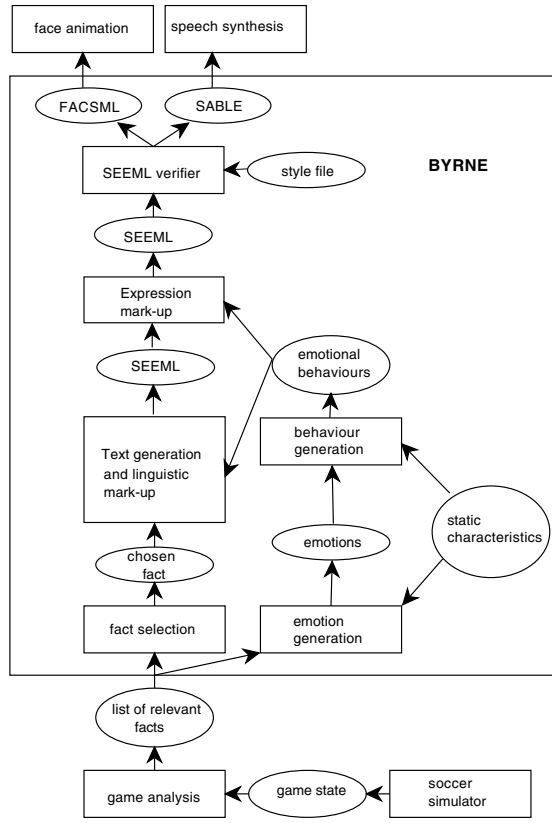

Fig. 1. The Byrne system architecture

so that non-experts in speech synthesis can mark up text in an intuitive manner which produces reasonable output from a variety of systems.

Although SABLE is in its early stages and does not yet have the richness required for the generation of affective speech (such as that described in [4]), it is a useful standard.

FACS FACS [5] stands for the Facial Action Coding System, and is a set of all the Action Units (AUs) which can be performed by the human face. It is often used as a way of coding the articulation of a facial animation [15]. For example, AU9 is known as the "nose wrinkler", and is based on the facial muscles Levator Labii Superioris and Alaeque Nasi. There are 46 AUs.

Although FACS is not an SGML-based mark-up language, it can be trivially converted into one by treating each AU as empty SGML element. Here we refer to such a mark-up as FACSML.

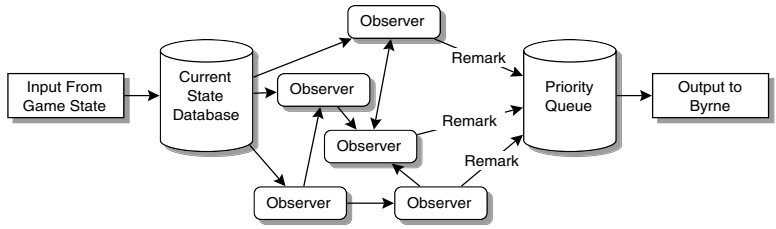

Fig. 2. The game analysis system.

4 Character architecture

Byrne is a system for expressive, entertaining commentary on a RoboCup Simulation League soccer game. Here we describe the Byrne character architecture (see Figure 1).

4.1 Input

Byrne can use any modular game analysis system as its input module (MIKE, for example). For RoboCup98, however, Byrne used a simple but effective play-by-play game analysis system designed for the competition (see Figure 2).

Byrne's input module listens for game state information reported from the RoboCup soccer monitor, and correspondingly updates a database of current state of play. The input module filters this information through *observers*, special objects which make abstract conclusions about the game, everything from ball ownership to noticing goals and keeping score to managing statistics. These conclusions take two forms. First, observers make many rudimentary *observations*, acknowledgement of changes in observed features which may or may not be very important. Many observers depend on both the state database and other observers' observations in order to make their own. Second, observers make *remarks*, occasional facts which could be commented on by Byrne.

Byrne's input module produces remarks much faster than Byrne is capable of saying them. To compensate for this, the input module feeds its remarks into a priority queue. Each remark has a *birthday* (the time when it was entered into the queue), a *deadline* (a time beyond which it is "old news"), and a *priority*. When Byrne requests a new fact to say, the queue returns one using a simple priority-scheduling algorithm. First, any fact in the queue whose deadline has past is deleted from the queue. Then the queue picks the fact **F** with the highest priority, and secondarily (to break ties) the earliest birthday. Finally, every fact with a birthday earlier than **F** is deleted from the queue. **F** is removed from the queue and returned to Byrne to comment on.

4.2 Emotion generation

The emotion generation module contains rules which generate simple *emotional structures*. These structures consist of:

- **a type**, e.g. *happiness, sadness*, etc. At present we are using Ekman's six basic emotions (*fear, anger, sadness, happiness, disgust* and *surprise*) [5], as research has shown that these are clearly and unambiguously expressible. We also include *interest* [1], as it is important in our domain of sports reporting, and is relatively easy to express in speech. In the future, however, we plan to have a hierarchy of emotions available, to allow for a richer range of resultant behaviours.
- **an intensity**, scored from 1 to 10. An emotion with intensity less than one is considered inactive, and is deleted from the emotion pool. Note that apparently opposite emotions (such as *happiness* and *sadness*) are not considered to be true opposites in Byrne. That is, a *sadness* structure is not merely a *happiness* structure with an intensity of -10. This is because *happiness* and *sadness* are not opposites in the sense that they cannot coexist, but only in that the behaviours they tend to inspire often conflict — that is, it is hard to express joy and sadness at the same time. Thus, this apparent conflict is resolved in the emotional behaviour rules, rather than in the emotional structures themselves. Moreover,emotion structures of the same type, but with different causes and/or targets, can coexist in the emotion pool.
- **a target** [optional]. Some emotions, such as *anger* and *interest*, are usually directed at some person or object. This is the **target** of that emotion.
- **a cause**. This is the fact about the world (in this case, the soccer game) which caused the emotion to come into being.
- **a decay function**. This is an equation describing how the intensity of the emotion decays over time, where time is given in seconds. Feelings which do not decay (e.g. a permanent dislike of a particular player) are not considered to be emotions for our purposes, and belong among the static characteristics of the character.

So, if a character is very sad about Team A having just scored, the relevant emotional structure might be:

(type:sadness, intensity:10, target:nil, cause:(scored team:a time:125) decay:1/t)

The intensity of this emotion would go down each second for ten seconds, then when it goes below one on the eleventh second, the emotional structure would be deleted from the emotion pool.

An emotion structure generation rule consists of a set of preconditions, which are to be filled by matching them on the currently true facts about the world

[1] For our purposes, "interest" is that emotion which at high intensity is called "excitement" and low intensity is called "boredom".

and about the character, and currently active emotion structures, the emotional structures to be added to the emotion pool, and the emotional structures to be removed. For example:

Preconditions:
(supports team: ?team)
(scores team: ?team)
Emotional structures to add:
(type: happiness intensity: 8 target: nil cause: (scores team: ?team) decay: 1/t)
Emotional structures to delete:
none

This rule indicates that, if the team that the commentator supports scores, a happiness structure should be added to the emotion pool.

There are only two ways for an emotion structure to be removed from the emotion pool: it can be explicitly removed by an emotion structure update rule, or its intensity can decay to below one, in which case it is automatically removed. In future, it might be necessary to develop a more sophisticated emotion maintenance system; however, since we have no ambitions to cognitive plausibility, we will only add such complications as necessary. We expect that this very simple emotional maintenance method will suffice for most of the situations a soccer commentator is likely to face.

Both emotion generation and behaviour generation are influenced by the **static characteristics** of the commentator character. This is a set of static facts about the character, such as his nationality, the team he supports, and so on. It is used to inform emotion and behaviour generation, allowing a character to react in accordance with his preferences and biases. For example, if a character supports the team which is winning, his emotional state is likely to be quite different that if he supports the losing team.

These emotion-generation rules can be arbitrarily complex, to take into account the context of both the character's static characteristics and the state of the world (in the case of soccer, the game).

4.3 Emotional behaviours

Emotion structures and static characteristics are preconditions to the activation of high-level emotion-expressing behaviours. These in turn decompose into lower-level behaviours. The lowest level behaviours specify how the text output by the text generation system is to be marked up.

Emotionally-motivated behaviours are organized in a hierarchy of mutually inconsistent groups. If two or more activated behaviours are inconsistent, the one with the highest activation level is performed. This will usually result in the strongest emotion being expressed; however, a behaviour which is motivated by several different emotions might win out over a behaviour motivated by one strong emotion.

It is entirely possible for mixed emotional expressions to be generated, as long as they are not inconsistent. For example, a happy and excited character might express excitement by raising the pitch of his voice and happiness by smiling. However, it is less likely that a character will have a way to express, say, happiness and sadness in a consistent manner.

4.4 Text generation

Character has a role to play in natural language generation. For example, a character from a particular country or region might use the dialect of that area, or a child character might use simpler vocabulary and grammar than an adult. The current emotional state of the character would also have an effect: an excited, angry character might use stronger language and shorter sentences than a calm happy one, for example. Loyall's work in the OZ project [12] discusses some of these issues.

Despite these possibilities for affective generation, text generation is an area in which Byrne does almost nothing of interest at present. It is done very simply through a set of templates. Each template has a set of preconditions which constrain the game situations they can be used to describe. If more than one template matches the chosen content, then the selection is based on how often and how recently the templates have been used.

Byrne's text generation module does not generate plain text, but rather text marked up with SEEML. Although the speech synthesis system we use can generate reasonable speech from plain text, it is helpful to retain some phrase structure and part of speech (POS) information from the natural language generation process to help the speech synthesis system to generate appropriate prosody.

Moreover, linguistic information embedded in the text also helps determine appropriate interruption points, should a more important fact need to be expressed. Here we assume that Byrne should finish the phrase it is currently uttering before it interrupts and starts a new utterance. This is a very simplistic approach, and may not be adequate.

Finally, linguistically-motivated facial gestures and speech intonation are now hard-coded into the templates. If the natural language generation system were more sophisticated, then a post-generation gesture and intonation system might be necessary, but with simple template generation this is the most effective method.

4.5 Expressive mark-up

SEEML[2] (the speech, expression and emotion mark-up language) is really just a slightly supplemented superset of three different mark-up systems, namely FACSML, SABLE and GDA. GDA is used to inform linguistically motivated expressive behaviours, and also to aid the speech synthesizer in generating appropriate prosody. FACSML is used to add facial behaviours, and SABLE is used to control the speech synthesis.

[2] SEEML is pronounced "seemly".

The expression mark-up module adds emotionally motivated mark-up to the already marked-up text from the text generation system. Conflicts are resolved in a simple (perhaps simplistic) manner. The combination rules are:

- Unless identical, tags are assumed to be independent. Any potential practical conflicts are left to be resolved by the speech synthesizer and/or facial animation systems.
- If two identical tags are assigned to a piece of text, the one with the smaller scope is assumed to be redundant, and removed.
- If two otherwise identical tags call for a change in some parameter, it is assumed that that change is additive.

4.6 SEEML verifier and style file

The SEEML verifier interprets SEEML tags in the context of a style file, adds time markers and lip synching information, and sends appropriate FACS to the facial animation system and SABLE (supplemented with phrase structure and POS tags) to the speech synthesis system.

Although sophisticated lip-synchronization algorithms have been developed (e.g. in [22]), they are not necessary for our purposes. Instead, we use a simple 'cartoon style' lip animation, which only shows the more obvious phoneme-viseme matches, as described in [15].

The style file contains speech and animation system specific interpretation rules. For example, it would determine the FACS which are to be used to indicate a **smile** for this particular face model, and the sound file to be used for a **hiccup**.

5 Implementation

In the first implementation, Byrne uses Franks and Takeuchi's facial animation system [19] and the Festival speech system [2]. We hope that the standardized mark-up of the output will allow the use of other systems as well.

The facial animation system is based on Waters' [22] polygon face model, and was implemented by Franks in C++ to run on a Silicon Graphics machine. It has a wide range of expressions, although the lip movement does not allow for sophisticated lip synching.

Festival is a concatenative speech synthesis system. It can generate speech in a number of different languages and voices, and the resulting speech is reasonably natural-sounding, although the user's control of the speech synthesis is quite coarse. Its other important feature for our purposes is its support of the Sable speech markup system.

Byrne itself is implemented in C++.

6 Future work

Although the emotional behaviours outlined above are very simple, this architecture allows for quite complex behaviours to be implemented. Maintaining

consistency and believability in these behaviours is a significant problem. We plan to develop a set of tools for character designers to help them in in this task.

Also, the current interruption mechanism is too simplistic, and sometimes results in clumsy commentary. We intend to devise a more sophisticated mechanism.

Although the character described throughout is a play-by-play commentator character, we hope to develop other characters for the soccer simulator domain, such as a coach or a soccer fan. We would also like to develop a colour commentator [3] to work with the play-by-play, which would necessitate introducing a turn-taking mechanism into the system.

7 Conclusion

In this paper we have motivated and described an architecture for a soccer commentator system, Byrne. Byrne generates emotional, expressive commentary of a RoboCup simulator league soccer game. It makes heavy use of inter-system standards, so that the system can take advantage of advances in speech synthesis, facial animation and natural language generation.

References

1. Elisabeth Andre, Gerd Herzog, and Thomas Rist. Generating multimedia presentations for robocup soccer games. Technical report, DFKI GmbH, German Research Center for Artificial Intelligence, D-66123 Saarbrucken, Germany, 1998.
2. Alan W. Black, Paul Taylor, and Richard Caley. *The Festival Speech Sythesis System*. CSTR, University of Edinburgh, 1.2 edition, September 1997.
3. Bruce Blumberg and Tinsley Galyean. Multi-level control for animated autonomous agents: Do the right thing... oh, not that... In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors*, pages 74–82. Springer-Verlag Lecture Notes in Artificial Intelligence, 1997.
4. Janet Cahn. Generating expression in sythesized speech. Master's thesis, Massachusetts Institute of Technology Media Laboratory, Boston, May 1989.
5. Paul Ekman and Erika L. Rosenberg, editors. *What the face reveals: Basic and applied studies of spontaneous expression using the facial action coding system*. Oxford University Press, 1997.
6. Athomas Goldberg. IMPROV: A system for real-time animation of behavior-based interactive synthetic actors. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors*, pages 58–73. Springer-Verlag Lecture Notes in Artificial Intelligence, 1997.
7. Charles Goldfarb. *The SGML Handbook*. Clarendon Press, 1991.
8. Barbara Hayes-Roth, Robert van Gent, and Daniel Huber. Acting in character. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors*, pages 92–112. Springer-Verlag Lecture Notes in Artificial Intelligence, 1997.
9. Keith Johnstone. *Impro*. Routledge Theatre Arts Books, 1992.

[3] A colour commentator provides background details on teams and players, such as statistics or amusing anecdotes.

10. Java speech markup language specification [0.5 beta]. Technical report, Sun Microsystems, 1997.
11. Hiroaki Kitano. Robocup. In Hiroaki Kitano, editor, *Proceedings of the IJCAI workshop on Entertainment and AI/ALife*, 1995.
12. A. Bryan Loyall. Some requirements and approaches for natural language in a believable agent. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors*, pages 113–119. Springer-Verlag Lecture Notes in Artificial Intelligence, 1997.
13. Katashi Nagao and Koiti Hasida. Automatic text summarization based on the global document annotation. Technical report, Sony Computer Science Laboratory, 1998.
14. Itsuki Noda. *Soccer Server Manual Rev. 2.00*, May 1997.
15. Frederic I Parke and Keith Waters. *Computer Facial Animation*. A K Peters Ltd, Wellesley, MA, 1996.
16. W. Scott Neal Reilly. *Believable social and emotional agents*. PhD thesis, School of Computer Science, Carnegie Mellon University, May 1996.
17. Draft specification for sable version 0.1. Technical report, The Sable Consortium, 1998.
18. R. Sproat, Paul Taylor, and Amy Isard. A markup language for text-to-speech synthesis. In *Proceedings of EUROSPEECH*, Rhodes, Greece, 1997.
19. Akikazu Takeuchi and Steven Franks. A rapid face construction lab. Technical Report SCSL-TR-92-010, Sony Computer Science Laboratory, Tokyo, Japan, May 1992.
20. Kumiko Tanaka-Ishii, Itsuki Noda, Ian Frank, Hideyuki Nakashima, Koiti Hasida, and Hitoshi Matsubara. MIKE: An automatic commentary system for soccer. Technical Report TR-97-29, Electrotechnical Laboratory, Machine Inference Group, Tsukuba, Japan, 1997.
21. Paul Taylor and Amy Isard. SSML: A speech synthesis markup language. *Speech communication*, 1997.
22. Keith Waters and Thomas M. Levergood. DECFace: An automatic lip-synchronization algorithm for sythetic faces. Technical Report CRL 93/4, Digital Cambridge Research Laboratory, September 1993.