

Three RoboCup Simulation League Commentator Systems

Elisabeth André*, Kim Binsted**, Kumiko Tanaka-Ishii+
Sean Luke++, Gerd Herzog*, Thomas Rist*

*DFKI, Germany, {andre,herzog,rist}@dfki.de

**Sony CSL, Japan, kimb@csl.sony.co.jp

+ETL, Japan, kumiko@etl.go.jp

++University of Maryland, USA, seanl@cs.umd.edu

Abstract

Three systems which generate real-time natural language commentary on the RoboCup simulation league are presented, and their similarities, differences and directions for the future discussed. Although they emphasize different aspects of the commentary problem, all three systems take simulator data as input, and generate appropriate, expressive, spoken commentary in real time.

Keywords: natural language generation, game analysis, face animation.

Introduction

Here we present three RoboCup simulation league commentator systems: Rocco, from DFKI; Byrne, from Sony CSL; and MIKE, from ETL. Together, the three systems won the scientific award at RoboCup'98 (Asada 1998) for making a significant and innovative contribution to RoboCup-related research.

Soccer is an interesting test domain because it provides a dynamic, real-time environment in which it is still relatively easy for tasks to be classified, monitored and assessed. Moreover, a commentary system has severe time restrictions imposed by the flow of the game, and is thus a good testbed for research into real-time systems. Finally, high-quality logs of simulated soccer games are already available and allow us to abstract from the intrinsically difficult task of low-level image analysis.

From Visual Data to Live Commentary

The automated generation of live reports on the basis of visual data constitutes a multi-stage transformation process. In the following, we describe how the maintainable subtasks transform the input into the final output.

The Input

All three commentary systems concentrate on the RoboCup simulator league, which involves software agents only (as opposed to the real robot leagues). Thus, the soccer games to be commented are not observed visually. Rather, all three systems obtain their

basic input data from the SOCCER SERVER (Kitano *et al.* 1997).

All three commentator systems use as input the same information that the monitor program receives for updating its visualizations. This information consists of:

- **player** location and orientation (for all players),
- **ball** location, and
- **game** score and play modes (such as throw-ins, goal kicks, *etc.*).

Game Analysis

Game analysis provides an interpretation of the input data and aims at recognizing conceptual units at a higher level of abstraction. The information units resulting from such an analysis encode a deeper understanding of the time-varying scene to be described. They include spatial relations for the explicit characterization of spatial arrangements of objects, as well as representations of recognized object movements. Scene interpretation is highly domain specific. Since we are dealing with soccer games, it is often possible to infer higher-level concepts from the observed motion patterns of the agents.

Topic Control and Content Selection

The next stage is to select the information to be communicated to the user. This selection depends on a number of constraints, such as:

- the game situation,
- the purpose of the comment,
- the user's information needs, and
- available presentation media.

As the commentary output is basically speech, we also have to consider that only one thing can be said at a time. In addition, the whole utterance should be consistent and well planned.

Natural Language Generation

Once the content of the next utterance is decided, the *text realization* is performed, which comprises grammatical encoding, linearization and inflection. Commentary is a real-time live report — information must

be communicated under time pressure and in a rapidly changing situation. The system should choose whether to make short telegram-style utterances or grammatically complete and explanatory utterances according to the situation. Also, it may sometimes be necessary for the commentator to interrupt itself. For example, if an important event (e.g., a goal kick) occurs, utterances should be interrupted to communicate the new event as soon as possible.

The Output

Finally, the natural-language utterances are piped to a speech synthesis module. To get more natural speech output, these systems do not rely on the default intonation of the speech synthesizer, but generate specific synthesizer instructions for expressing the corresponding emotions. These emotions, whether hardwired into the templates or generated in accordance with an emotional model and the state of the game, can also be shown in the facial expressions of an animated commentator character.

While Mike and Rocco produce disembodied speech, Byrne makes use of a face as an additional means of communication.

Rocco

The ROCCO commentator system is a reincarnation of an early research prototype - called SOCCER - which was built by André, Herzog and Rist in the late 80's for the automated interpretation and natural language description of time-varying scenes (André, Herzog, & Rist 1988). At that time, short sections of video recordings of soccer games were chosen as a major application domain since they offered interesting possibilities for the automatic interpretation of visually observed motion patterns in a restricted domain. Later on, the work on incremental scene interpretation (Herzog *et al.* 1989; Herzog & Wazinski 1994; Herzog & Rohr 1995) was combined with an approach for plan-based multimedia presentation design (André *et al.* 1993; André & Rist 1990) in order to move towards the generation of various multimedia reports, such as TV-style reports and illustrated newspaper articles (André, Herzog, & Rist 1994). Part of this extension was a generic architecture for multimedia reporting systems which describes the representation formats and the multi-stage transformation process from visual data to multimedia reports. Both systems, the early SOCCER as well as the new ROCCO system, can be seen as instantiations of this architecture. In the following, we briefly sketch ROCCO's main components and the underlying approaches for incremental event recognition and report generation.

In contrast to the early SOCCER system, ROCCO does not have to tackle the intrinsically difficult task of automatic image analysis. Instead of using real image sequences and starting from raw video material, it relies on the real-time game log provided by the SOCCER SERVER. Based on these data, ROCCO's incremental

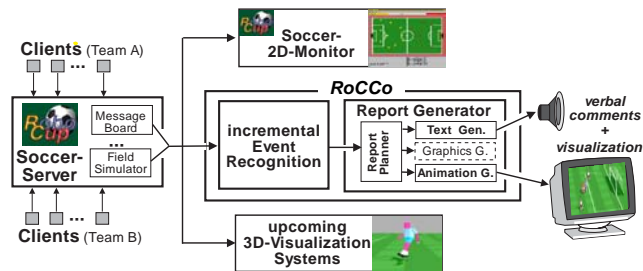


Figure 1: Connecting SOCCER SERVER and ROCCO

event recognition component performs a higher level analysis of the scene under consideration. The representation and automatic recognition of events is inspired by the generic approach described in (Herzog & Wazinski 1994), which has been adopted in ROCCO according to the practical needs of the application context. Declarative event concepts represent a priori knowledge about typical occurrences in a scene. These event concepts are organized into an abstraction hierarchy. Event concepts, such as locomotions or kicks are found at lower levels of the hierarchy since they are derived through the principle of specialization and temporal decomposition of more complex event concepts such as a ball transfer or an attack. Simple recognition automata, each of which corresponds to a specific concept definition, are used to recognize events on the basis of the underlying scene data.

The recognized occurrences along with the original scene data form the input for report generation. To meet the specific requirements of a live report, ROCCO's report planner relies on an incremental discourse planning mechanism. Assuming a discrete time model, at each increment of a time counter the system decides which events have to be communicated next. Thereby, it considers the salience of events and the time that has passed since their occurrence. If there are no interesting events, the system randomly selects background information from a database.

The text generator is responsible for transforming the selected information into spoken utterances. Since it is rather tedious to specify soccer slang expressions in existing grammar formalisms, we decided to use a template-based generator instead of fully-fledged natural-language generation components as in the earlier SOCCER system. That is, language is generated by selecting templates consisting of strings and variables that will be instantiated with natural-language references to objects delivered by a nominal-phrase generator. To obtain a rich repertoire of templates, 13.5 hours of TV soccer reports in English have been transcribed and annotated. Templates are selected considering parameters, such as available time, bias, and report style. For instance, short templates, such as "Meier again", are preferred if the system is under time pressure. Furthermore, ROCCO maintains a discourse

history to avoid repetitions and to track the center of attention. For instance, a phrase like "Now Miller" should not be uttered if Miller is already topicalized. For the synthesis of spoken utterances, ROCCO relies on the TRUETALK (Entropic Research Laboratory Inc) text-to-speech software. To produce more lively reports, it annotates the computed strings of words with intonational markings considering the syntactic structure and the content of an utterance as well as the speaker's emotions. Currently, it mainly varies pitch accent, pitch range and speed. For instance, excitement is expressed by a higher talking speed and pitch range.

Fig. 2 shows a screenshot of the initial Java-based ROCCO prototype which was taken during a typical session with the system. The monitor program provided with the RoboCup simulation environment is used to play back the previously recorded match in the upper right window. The text output window on the left-hand side contains a transcript of the spoken messages that have been generated for the sample scene. ROCCO does not always generate grammatically complete sentences but primarily produces short telegram-style utterances which are more typical of live TV reports. The testbed character of the ROCCO system provides the possibility of experimenting with various generation parameters, such as language style, which may be set in the lower right window.



Figure 2: The basic windows of the initial ROCCO system

Byrne

Now we present early work on an animated talking head commentary system called **Byrne**. This system takes the output from the RoboCup soccer simulator, and generates appropriate affective speech and facial expressions, based on the character's personality, emotional state, and the state of play (Binsted 1998).

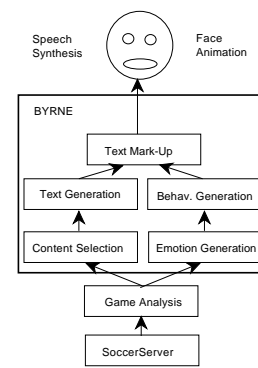


Figure 3: The Byrne system architecture



Figure 4: Byrne sneering.

Byrne can use any modular game analysis system as its input module. For RoboCup98, however, Byrne used a simple but effective play-by-play game analysis system designed for the competition. This 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.

The information provided by the analysis system also drives the emotional model. The emotion generation module contains rules which generate simple *emotional structures*. These structures consist of: **a type**, e.g. *happiness*, *sadness*, etc.; **an intensity**, scored from 1 to 10; **a target** [optional]; **a cause**, i.e. the fact about the world which caused the emotion to come into being; and **a decay function**, describing how the intensity of

the emotion decays over time.

An emotion structure generation rule consists of a set of preconditions, the emotional structures to be added to the emotion pool, and the emotional structures to be removed. The preconditions are filled by matching on the currently true facts about the world and about the character.

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/her nationality, the team s/he supports, and so on. It is used to inform emotion and behaviour generation, allowing a character to react in accordance with his/her preferences and biases. For example, if a character supports the team which is winning, his/her emotional state is likely to be quite different that if s/he supports the losing team.

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.

Emotions are expressed by adding mark-up to already-generated text. Text generation 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 (see below). Linguistically-motivated facial gestures and speech intonation are currently hard-coded into the templates.

SEEML (the Speech, Expression and Emotion Mark-up Language) is a slightly supplemented superset of three different mark-up systems, namely FACSML (a variant on FACS (Ekman & Friesen 1978)), SABLE (The Sable Consortium 1998) and GDA (Nagao & Hasida 1998). 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.

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. Unless identical, tags are assumed to be independent. If two identical tags are assigned to a piece of text, the one with the

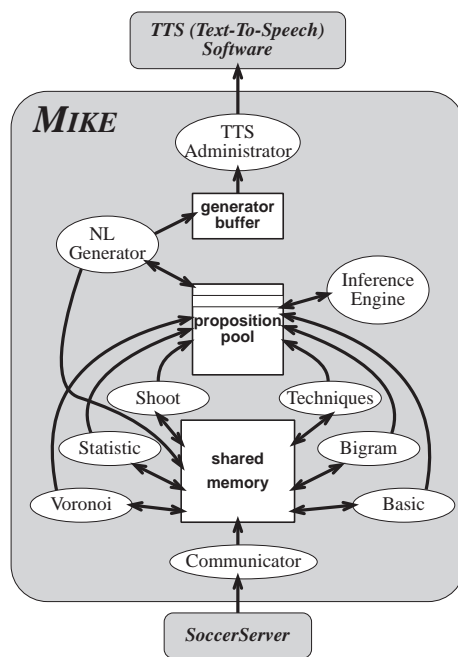


Figure 5: MIKE's structure

smaller scope is assumed to be redundant, and removed. Finally, if two otherwise identical tags call for a change in some parameter, it is assumed that that change is additive.

The marked-up text is then sent to the SEEML parser. The parser interprets SEEML tags in the context of a style file, adds time markers and lip syncing information, and sends appropriate FACS to the facial animation system and SABLE to the speech synthesis system. The result is an emotional, expressive talking-head commentary on a RoboCup simulation league soccer game.

MIKE

MIKE¹ is an automatic real time commentary system capable of producing output in English, Japanese, and French.

One of our contributions is to demonstrate that a collection of concurrently running analysis modules can be used to follow and interpret the actions of multiple agents (Tanaka-Ishii *et al.* 1998). MIKE uses six Soccer Analyzer modules, three of which carry out high-level tasks. Notably, these modules demonstrate the general applicability of analyzing the *focus* of a multi-agent system, and of examining the *territories* established by individual agents.

Another technical challenge involved in live commentary is the real time selection of content to describe the complex, rapidly unfolding situation (Tanaka-Ishii, Hasida, & Noda 1998). Soccer is a multi-agent game

¹The name MIKE is an acronym for 'Multi-agent Interactions Knowledgeably Explained'.

- **Explanation of complex events:** This concerns form changes, position changes, and advanced plays.
- **Evaluation of team plays:** This concerns average forms, forms at a certain moment, players' locations, indications of the active or problematic players, winning passwork patterns, and wasteful movements.
- **Suggestions for improving play:** These concern loose defense areas and better locations for inactive players.
- **Predictions:** These concern passes, game results, and shots at goal.
- **Set pieces:** These concern goal kicks, throw ins, kick offs, corner kicks, and free kicks.
- **Passwork:** This tracks basic ball-by-ball plays.

Figure 6: MIKE's commentary repertoire

Red3 collects the ball from Red4, Red3, Red-Team, wonderful goal! 2 to 2! Red3's great center shot! Equal! The Red-Team's formation is now breaking through enemy line from center, The Red-Team's counter attack (Red4 near at the center line made a long pass towards Red3 near the goal and he made a shot very swiftly.), Red3's goal! Kick off, Yellow-Team, Red1 is very active because, Red1 always takes good positions, Second half of Robocup-97 quarterfinal(Some background is described while the ball is in the mid field.) . Left is Ohta Team, Japan, Right is Humboldt, Germany, Red1 takes the ball, bad pass, (Yellow team's play after kick off was interrupted by Red team) Interception by the Yellow-Team, Wonderful dribble, Yellow2, Yellow2 (Yellow6 approaches Yellow2 for guard), Yellow6's pass, A pass through the opponents' defense, Red6 can take the ball,because, Yellow6 is being marked by Red6, The Red-Team's counter attack, The Red-Team's formation is (system's interruption), Yellow5, Back pass of Yellow10, Wonderful pass,

Figure 7: Example of MIKE's commentary

in which various events happen simultaneously in the field. In order to weave a consistent and informative commentary on such a subject, an importance score is put on each fragment of commentary that intuitively captures the amount of information communicated to the audience. The inference and the content selection modules are both controlled by such importance scores.

From the input sent by the Soccer Server, MIKE creates a commentary that can consist of any combination of the possible repertoire of remarks, as shown in Figure 6. An English example of MIKE's commentary is shown in Figure 7. This commentary generation is coordinated by the architecture shown in Figure 5, where the ovals represent processes and the rectangles represent data.

There are six Soccer Analyzer Modules, of which three analyze basic events (shown in the figure as the 'Basic', 'Techniques', and 'Shoot' processes), and the other three carry out more high-level analysis (shown as the 'Bigram', 'Voronoi', and 'Statistic' processes). These six processes analyze the information posted to the shared memory by the Communicator, communicate with each other via the shared memory, and also post propositions to the proposition pool.

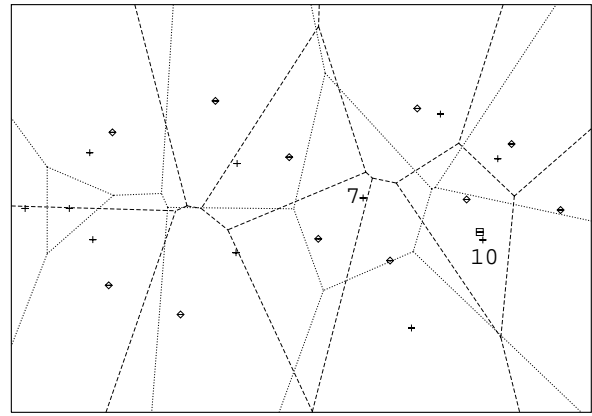


Figure 8: An example of a Voronoi diagram

The Bigram module follows and analyzes the ball plays as a first order Markov chain that is a *focus* point in soccer game. A 24×24 ball play transition matrix (22 players and 2 goals) is automatically formed during the game and used to describe the activity of each player and to identify successful passwork patterns. Using this matrix, MIKE calculates and examines the players' ball play performance such as pass success rates, winning passwork patterns, number of shoots.

The Voronoi module calculates Voronoi diagrams for each team every 100ms. Using these partitions, MIKE can determine the defensive areas covered by players and also assess overall positioning. Figure 8 shows an example of such a Voronoi diagram ('+' and '◇' indicate players of each team, '⊠' shows the ball.).

In the future, these rich state analysis modules will be separated from the rest of the system into a 'preMIKE' module, so that simulation soccer teams or other commentary systems can dynamically refer to the analysis results.

All inter-process communication is done via the handling of an internal representation of commentary fragments, represented as *propositions*. These proposition consists of a tag and some attributes. For example, a kick by player No.5 is represented as (Kick 5), where Kick is the tag and 5 is the attribute.

To establish the relative significance of events, importance scores are put on propositions. After being initialized in the Analyzer Module, the score decreases over time while it remains in the pool waiting to be uttered. When the importance score of a proposition reaches zero, it is deleted from the pool.

Propositions deposited in the pool are often too low-level to be directly used to generate commentary. The Inference Engine Module processes the propositions in the pool with a collection of over 100 forward chaining inference rules. The rules can be categorized into four classes, based on the relation of consequences to their

	Analysis	Methods for Topic Control	NL Generation	Output
Rocco	Recognition automata	Ranking based on salience, information gain and age of events	Parameterized template selection + NP generation in real-time	Expressive speech
Byrne	‘Observers’ recognize events and states	Prioritized by Birthday, Deadline and Priority scores	Templates, marked up for expression and interruption in real-time	Expressive speech and facial animation
MIKE	Events and states (bigrams, voronoi)	Importance scores and inference	Templates, Interruption, abbreviation using importance scores	Expressive speech

Figure 9: A comparison of the features of the three commentator systems.

antecedents: logical consequences, logical subsumption, second order relations and state change. For example, (`HighPassSuccessRate player`) (`PassPattern player Goal`) \rightarrow (`active player`) is a rule to make a logical consequence.

Finally, the Natural Language Generator selects the proposition from the proposition pool that best fits the current state of the game, and then translates the proposition into natural language. It broadcasts the current subject to the analyzers, so that they assign higher initial importance values to propositions with related subjects. Content selection is made with the goal of maximizing the total gain of importance scores during the game. Thus, the content selection is integrated in the surface realization module, which accounts for interruption, abbreviation, and so on.

Discussion and Future Work

Even though our focus has been on the automatic description of soccer matches, none of the presented approaches is restricted to this class of application. For instance, the group at DFKI originally addressed the dynamic description of traffic scenes (cf. (Herzog & Wazinski 1994)). To port our approach to the domain of soccer, we didn’t have to change a single line in the processing algorithms. Only our declarative knowledge sources, such as the event recognition automata and the natural-language templates, had to be adapted.

At Sony CSL, we are currently looking at adapting our systems to commentate other domains, in particular, other sports (e.g. baseball, American football). Also, in real sports commentary, commentators usually work in pairs. One is the *play-by-play* commentator, while the other provides *colour* (i.e. background information and statistics about the players and teams). We are interested in having our commentator systems mimic these roles, taking turns providing relevant information in context.

DFKI is exploring new forms of commentary as well. We are currently working on a 3D visualization component to enable situated reports from the perspective of a particular player.

As an application of MIKE, ETL is now developing a real-time navigation system for pedestrians, especially blind people, and also for automobiles. This system,

like MIKE, senses the position of the target to be navigated, makes analysis of the situation using static information as maps, controls topic, and then outputs the navigation as expressive speech.

Conclusions

Although there are some differences in emphasis, all three systems provide real-time, expressive commentary on the RoboCup simulation league. All were demonstrated at RoboCup’98, and we hope that improved versions will be shown at RoboCup’99 in Stockholm. The success of these systems shows that RoboCup is not just a robot competition — it is a challenging domain for a wide range of research areas, including those related to real-time natural language commentary generation.

Acknowledgements

We would like to thank Dirk Völz for his work on the implementation of the ROCCO system.

References

- André, E., and Rist, T. 1990. Towards a plan-based synthesis of illustrated documents. In *Proceedings of the 9th ECAI*, 25–30.
- André, E.; Finkler, W.; Graf, W.; Rist, T.; Schauder, A.; and Wahlster, W. 1993. Wip: The automatic synthesis of multimodal presentations. In Maybury, M. T., ed., *Intelligent Multimedia Interfaces*. Menlo Park, CA: AAAI Press. 75–93.
- André, E.; Herzog, G.; and Rist, T. 1988. On the simultaneous interpretation of real world image sequences and their natural language description: The system SOCCER. In *Proceedings of the 8th ECAI*, 449–454.
- André, E.; Herzog, G.; and Rist, T. 1994. Multimedia presentation of interpreted visual data. In McKeivitt, P., ed., *Proceedings of AAAI-94 Workshop on “Integration of Natural Language and Vision Processing”*, 74–82. Also available as Report no. 103, SFB 314 – Project VITRA, Universität des Saarlandes, Saarbrücken, Germany.
- Asada, M., ed. 1998. *RoboCup-98: Robot Soccer World Cup II*.

- Binsted, K. 1998. Character design for soccer commentary. In *Proceedings of the second RoboCup workshop*, 23–35.
- Ekman, P., and Friesen, W. V. 1978. *Manual for the Facial Action Coding System*. Palo Alto, CA: Consulting Psychologists Press, Inc.
- Entropic Research Laboratory Inc. TrueTalk reference and programmers' manual, version 2.0. Product description available at <http://www.entropic.com/>.
- Herzog, G., and Rohr, K. 1995. Integrating vision and language: Towards automatic description of human movements. In Wachsmuth, I.; Rollinger, C.-R.; ; and Brauer, W., eds., *KI-95: Advances in Artificial Intelligence. 19th Annual German Conference on Artificial Intelligence*, 257–268. Berlin, Heidelberg: Springer.
- Herzog, G., and Wazinski, P. 1994. Visual TRANslator: Linking perceptions and natural language descriptions. *AI Review* 8(2/3):175–187.
- Herzog, G.; Sung, C.-K.; André, E.; Enkelmann, W.; Nagel, H.-H.; Rist, T.; Wahlster, W.; and Zimmermann, G. 1989. Incremental natural language description of dynamic imagery. In *Wissensbasierte Systeme. 3. Int. GI-Kongreß*. Berlin, Heidelberg: Springer. 153–162.
- Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawa, E.; and Matsubara, H. 1997. RoboCup: A challenge problem for ai. *AI Magazine* 18(1):73–85.
- Nagao, K., and Hasida, K. 1998. Automatic text summarization based on the global document annotation. Technical report, Sony Computer Science Laboratory.
- Tanaka-Ishii, K.; Noda, I.; Frank, I.; Nakashima, H.; Hasida, K.; and Matsubara, H. 1998. Mike: An automatic commentary system for soccer. In *Proceedings of ICMAS-98*.
- Tanaka-Ishii, K.; Hasida, K.; and Noda, I. 1998. Reactive content selection in the generation of real-time soccer commentary. In *Proceedings of COLING-98*.
- The Sable Consortium. 1998. Draft specification for sable version 0.1. Technical report.