Reactive Content Selection in the Generation of Real-time Soccer Commentary

Kumiko TANAKA-Ishii and **Kôiti HASIDA** and **Itsuki NODA**
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan.

Abstract

MIKE is an automatic commentary system that generates a commentary of a simulated soccer game in English, French, or Japanese.

One of the major technical challenges involved in live sports commentary is the reactive selection of content to describe complex, rapidly unfolding situation. To address this challenge, MIKE employs importance scores that intuitively capture the amount of information communicated to the audience. We describe how a principle of maximizing the total gain of importance scores during a game can be used to incorporate content selection into the surface generation module, thus accounting for issues such as interruption and abbreviation.

Sample commentaries produced by MIKE are presented and used to evaluate different methods for content selection and generation in terms of efficiency of communication.

1 Introduction

Timeliness, or reactivity, plays an important role in actual language use. An expression should not only be appropriately planned to communicate relevant content, but should also be uttered at the right moment to describe the action and further to carry on the discourse smoothly. Content selection and its generation are inseparable here. For example, people often start talking before knowing all that they want to say. It is also relatively common to fill gaps in commentary by describing what was true in the past. An extreme instance is when an utterance needs to be interrupted to describe a more important event that suddenly occurs.

It might be expected that dialogue systems should have addressed such real-time issues, but in fact these studies appear to have been much more focused on content planning. The reason for this lies in the nature of dialogue. Although many human-human conversations involve a lot of time pressure, slower conversations can also be successful provided the planning is sufficiently incorporated. For example, even if one conversation participant spends time before taking a turn, the conversation partner can just wait until hearing a contribution.

In contrast, reactivity is inevitable in live commentary generation, because the complexity and the rapid flow of the situation severely restrict what to be said, and when. If too much time is spent thinking, the situation will unfold quickly into another phase and important events will not be mentioned at the right time.

MIKE is an automatic narration system that generates spoken live commentary of a simulated soccer game in English, French, or Japanese. We chose the game of soccer firstly because it is a multi-agent game in which various events happen simultaneously in the field. Thus, it is a suitable domain to study real-time content selection among many heterogeneous facts. A second reason for choosing soccer is that detailed, high-quality logs of simulated soccer games are available on a real-time basis from Soccer Server(Noda and Matsubara, 1996), the official soccer simulation system for the RoboCup (Robotic Soccer World Cup) initiative.

The rest of the paper proceeds as follows. First, we describe our principle for real time content selection and explain its background. Then, after briefly explaining MIKE's overall design, §4 explains how our principles are realized within our implementation. §6 discusses some related works, and §5 presents some actual output by MIKE and evaluates it in terms of efficiency of communication.

2 Principles of Content Selection in the Real Time Discourse

2.1 Maximization of Total Information

A discourse is most effective when the amount of information transmitted to the listener is maximal.

In the case of making discourse about a static subject whose situation does not change, the most important contents can be selected and described in the given time.

In the case of making discourse on a dynamic subject, however, content selection suddenly becomes very complex. Above all, the importance of the contents changes according to the dynamic discourse topic, and also according to the dynamic situation of the subject. Additionally, past events become less important with time. Under this condition, the basic function of content selection is to choose the most important content at any given time. This control, however, is not enough, because any content will take time to be uttered and during that time, the situation of the subject might change rapidly. Therefore, it should always be possible to change or rearrange the content being uttered.

Examples of such rearrangements are:

- **interruption.** When the situation of the subject changes suddenly to a new one, more information can be given by rejecting the current utterance and switching to new one.
- **abbreviation.** When many important facts arise, the total information can be augmented by referring to each facts quickly by abbreviating each one.
- **repetition.** When nothing new comes up in the subject, the important facts already uttered can be repeated to reinforce the information given to the listener.

As a consequence, creating a system which involves real time discourse concerns 1.assessing the dynamic importance of contents, 2.controlling the content selection with this importance so that the total information becomes maximal using the rearrangement functions.

In §4, we discuss how we implemented these principles in MIKE to produce a real time narration.

2.2 *What*, *How* and *When-to-Say*

The previous section pointed out that contents should be uttered at the right time; that is, real time discourse systems should effectively address the problem of *when-to-say* any piece of information. However, in MIKE we have only an implicit model of *when-to-say*. Rather, a collection of game analysis modules and inference rules first suggest the possible comments that can be made (*what-to-say*). Then, an NL-generation module decides which of these comments to say (again *what-to-say*), and also how it should be realised (*how-to-say*). This *how-to-say* process takes into account issues such as the rearrangements described in the previous section.

In traditional language generation research, the relationship between the *what-to-say* aspect (planning) and the *how-to-say* aspect (surface generation) has been widely discussed (Appelt, 1982) (Hovy, 1988). One viewpoint is that, for designing natural language systems, it is better to realize *what-to-say* and *how-to-say* as separate modules. However, in MIKE we found that the time pressure in the domain makes it difficult to separate *what-to-say* and *how-to-say* in this way. Our NL generator decides both on *what-to-say* and *how-to-say* because the rearrangements made when deciding how to realize a piece of information directly affect the importance of the remaining unuttered comments. To separate these processes cause significant time delays that would not be tolerable in our time-critical domain.

- **Explanation of complex events** concern form changes, position change, and advanced plays.
- **Evaluation of team plays** concern average forms, forms at a certain moment, players' location, indication of the active or problematic players, winning passwork patterns, wasteful movements.
- **Suggestions for improving play** concern loose defense areas, and better locations for inactive players.
- **Predictions** concern pass, game result, and shots at goal.
- **Set pieces** concern goal kicks, throw ins, kick offs, corner kicks, and free kicks.
- **Passworks** track basic ball-by-ball plays.

Figure 1: MIKE's repertoire of statements

3 Brief Description of MIKE's Design

A detailed description, of MIKE, especially its soccer game analysis capabilities can be found in (Tanaka-Ishii et al., 1998). Here we simply give a brief overview.

3.1 MIKE's Structure

MIKE, 'Multi-agent Interactions Knowledgeably Explained', is designed to produce simultaneous commentary for the Soccer Server, originally proposed as a standard evaluation method for multi-agent systems(Noda and Matsubara, 1996). The Soccer Server provides a real-time game log[1] of a very high quality, sending information on the positions of the players and the ball to a monitoring program every 100msec. Specifically, this information consists of:

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

From this low-level input, the current implementation of MIKE can generate the range of comments shown in Figure 1.

[1] The simulator and the game logs are available at `http://ci.etl.go.jp/~noda/soccer/server`.

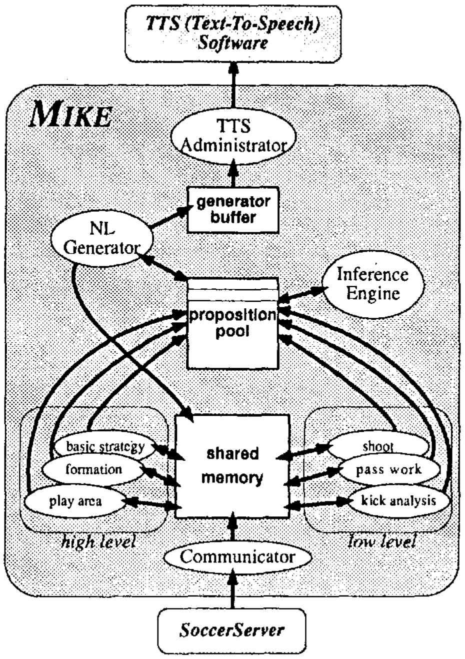

Figure 2: MIKE's structure

Table 1: Examples of *Propositions*, the internal fragments of commentary

	Local	Global
Event	Kick	ChangeForm
	Pass	
	Dribble	SideChange
	ShootPredict	
State	Mark	TeamPassSuccessRate
	PlayerPassSuccessRate	AveragePassDistance
	ProblematicPlayer	Score
	PlayerActive	Time

MIKE's architecture — a role-sharing multi-agent system [2] — is shown in Figure 2. Here, the ovals represent concurrently running modules and the rectangles represent data.

All communication among modules is mediated by the internal symbolic representation of commentary fragments, which we call *propositions*. A proposition is represented with 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. So far, MIKE has around 80 sorts of tags, categorized in two ways: as being local or global and as being state-based or event-based. Table 1 shows some examples of categorized proposition tags.

- **Logical consequences:**

  (PassSuccessRate *player percentage*)
  (PassPattern *player* Goal)
  → (active *player*)

- **Logical subsumption:**

  (Pass *player1 player2*)(Kick *player1*)
  → (Delete @2)

- **State change:**

  (Form *team form1*)(Form *team form2*)
  → (Delete earlier-prop)

- **Second order relation:**

  (PassSuccessRate *player percentage*)
  (PlayerOnVoronoiLine *playr*) →
  (Reason @1 @2)

Figure 3: Categories and examples of inference rules

Some of the important modules in MIKE's architecture can be summarized as follows.

There are six **Soccer Analyzers** that try to interpret the game. Three of these analyze events (shown in the figure as the 'kick analysis', 'pass work', and 'shoot' modules). The other three carry out state-based analysis (shown as the 'basic strategy', 'formation', and 'play area' modules). The modules analyze the data from the Soccer Server, communicate with each other via the shared memory, and then post the results as propositions into the Pool.

The **Real Time Inference Engine** processes the propositions. Prpositions deposited in the Pool are bare facts and are often too detailed to be used as comments. MIKE therefore uses forward chaining rules of the form

precedents→antecedents

to draw further inferences. The types of rules used for this process are shown in Figure 3. Currently, MIKE has about 110 such rules.

The **Natural Language Generator** selects the proposition from the Pool that best fits the current state of the game (considering both the situation on the field and the comment currently being made). It then translates the proposition into NL. So far, MIKE just carries out this final step with the simple mechanism of template-matching. Several templates are prepared for each proposition tag, and the out-

[2] In natural language processing, the multi-agent approach dates back to Hearsay-II (Erman et al., 1980), which was the first to use the blackboard architecture. The core organization of MIKE, however, is more akin to a subsumption architecture (Brooks, 1991), because the agents are regarded as behavior modules which are both directly connected to the external environment (through sensor readings from the shared memory) and can directly produce system behavior (by suggesting commentary). However, MIKE does not exactly fit the subsumption architecture model because the agents can also communicate with each other: there are some portions of the shared memory that are global and some that are exported to only a limited number of agents. This division of shared memory leads to more possibilities for inter-agent communication.

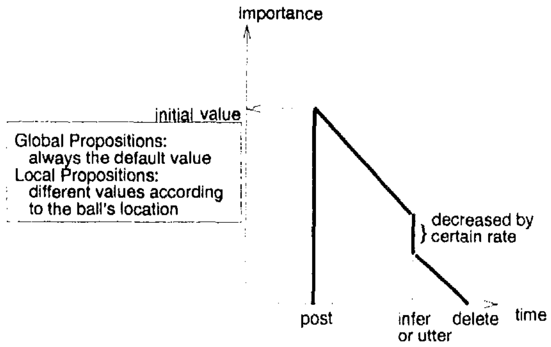

Figure 4: An example transformation of importance of a proposition

put can be is in English, French or Japanese.

To produce speech, MIKE uses off the shelf text-to-speech software. English is produced by Dectalk(DEC, 1994), French by Proverbe Speech Engine Unit(Elan, 1997), Japanese by Fujitsu Japanese Synthesizer(Fujitsu, 1995).

4 Implementation of Content Selection

4.1 Importance of a Proposition

The Soccer Analyzers attach an importance score to a proposition, which intuitively captures the amount of information that the proposition would transmit to an audience.

The importance score of a proposition is planned to change over time as follows (Figure 4). After being posted to the Pool, 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. This decrease in importance models the way that an event's relevance decreases as the game progresses.

The rate at which importance scores decrease can be modeled by any monotonic function. For simplicity, MIKE's function is currently linear. Since it seems sensible that local propositions should lose their score more quickly than global ones, several functions with different slopes are used, depending on the degree to which a proposition can be considered local or global. When a proposition is used for utterance or inference, the score is reduced in order to avoid the redundant use of the same proposition, but not set to zero, thus leaving a small chance for other inferences.

There is also an initialization process for the importance scores as follows. First, to reflect the situation of the game, the local propositions are modified by a multiplicative factor depending on the state of the game. This factor is designed so that local propositions are more important when the ball is near the goal. Global propositions are always initialized with the default value.

Secondly, to reflect the topic of the discourse, MIKE has a feedback control which enables each Soccer Analyzer module to take into account MIKE's past and present utterances. The NL generator broadcasts the current subject to the agents and they assign greater initial importance scores to propositions with related subjects. For example, when MIKE is talking about player No.5, the Analyzers assign a higher importance to propositions relating to this player No.5.

4.2 Maximization of the Importance Score

As the importance score is designed to intuitively reflect the information transmitted to the audience, the natural application of our content selection principles described in §2 is simply to attempt to maximize the total importance of all the propositions that are selected for utterance.

MIKE has the very basic function of uttering the most important content at any given time. That is, MIKE repeatedly selects the proposition with the largest importance score in the Pool.

The NL Generator translates the selected proposition into a natural language expression and sends it to the TTS-administrator module. Then the NL Generator has to wait until the Text-to-Speech software finishes the utterance before sending out the next expression. During this time lag, however, the game situation might rapidly unfold and numerous further propositions may be posted to the Pool. It is to cope with this time lag that MIKE implements a alternative function, that allows a more flexible selection of propositions by modeling the processes of interruption, abbreviation, and repetition,

Interruption

If a proposition with a much larger importance score than the one currently being uttered is inserted into the Pool, the total importance score may become larger by immediately interrupting the current utterance and switching to the new one. For example, the left of Figure 5 shows (solid line) the change of the importance score with time when an interruption takes place (the dotted line represents the importance score without interruption). The left part of the solid line is lower than the dotted, because the first utterance conveys less of its importance score (information) when it is not completely uttered. The right part of the dotted line is lower than that of the solid, because the importance of the second utterance decreases over time when waiting

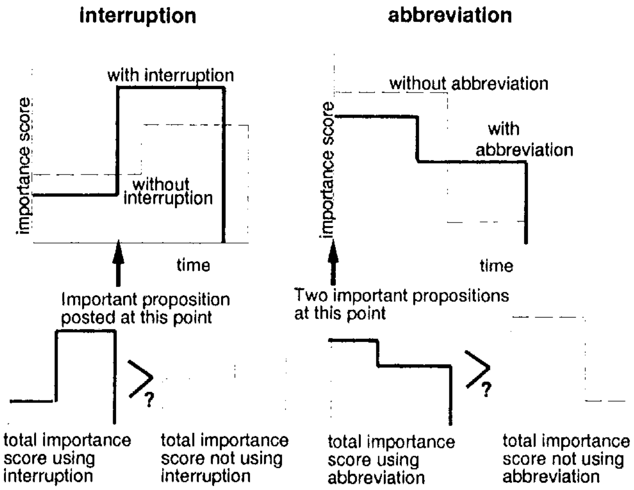
interruption
abbreviation
with interruption
without interruption
importance score
time
Important proposition posted at this point
without abbreviation
with abbreviation
importance score
time
Two important propositions at this point
?
?
total importance score using interruption
total importance score not using interruption
total importance score using abbreviation
total importance score not using abbreviation

Figure 5: Change of importance score on interruption and abbreviation

to be selected.

Thus, the sum of the importance of the uttered propositions can no longer be used to access the system's performance. Instead, the area between the lines and the horizontal axis indicates the total importance score over time. Whether or not to make interruption should be decided by comparing two areas made by the solid and dotted, and the larger area size is the total importance score gain. Further, this selection decides what to be said and how at the same time.

Note that interruptions raise the importance score gain by reacting sharply to the sudden increase of the importance score.

Abbreviation

If the two most important propositions in the Pool are of similar importance, it is possible that the amount of communicated information could be maximized by quickly uttering the most important proposition and then moving on to the second before loses importance due to some development of the game situation. In the Figure 5, we have illustrated this in the same way we did for the case of interruption. The left hand side of the solid line is lower than that of the dotted because an abbreviated utterance (which might not be grammatically correct, or whose context might not be fully given) transmits less information than a more complete utterance. As the second proposition can be uttered before losing its importance score, however, the right hand part of the solid line is higher than that of the dotted. As before, the benefits or otherwise of this modification should be decided by comparing with the two areas made by the solid and the dotted line with the horizontal axis. Again, this selection decides *how* and *what-to-say* at the same point.

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 quater-final*(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 Read 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 6: Example of Mike's commentary of a quater-final from RoboCup'97

In this case we would hope that abbreviations raise the importance score by smoothing sudden decreases of the importance scores posted to the Pool.

Repetition

Whenever a proposition is selected to be uttered, its importance value is decreased. It is also marked as having been uttered, to prevent its re-use. However, sometimes it can happen that the remaining unuttered propositions in the Pool have much smaller values than any of those that have already been selected. In this case, we investigate the effects of allowing previously uttered propositions to be repeated.

5 Evaluation

5.1 Output Example

An example of Mike's commentary (when employing interruption, abbreviation and repetition) is shown in Figure 6. In practice, this output is designed to accompany a visual game, but it is impractical to reproduce here enough screen-shots to describe the course of the play. We have therefore instead included some context and further explanations in parentheses. This particular commentary

赤4番のプレー、赤4番から赤3番、　赤3番、赤チームの得点です、2点です、2対2、赤3番のセンターからのシュート！同点！赤3番がゴール、赤チームは中央突破型です、黄色チームの速攻プレー、ロボカップ'97準々決勝後半戦です。黄色は太田チーム、日本、赤はフンボルト、ドイツ赤1番が取れそう、黄色チーム　、黄色5番の方向、赤6番が取れそう、なぜなら、赤6番は黄色6番をマークしています、赤5番が取れそう、赤チームのキックイン、赤8番、赤8番、黄色8番です、今度もまたシュートにつなげられるでしょうか、

Figure 7: Japanese output

Rouge4, Rouge4, Balle du Rouge4 au Rouge3, Rouge3, 2e but. Score de 2à 2. Tir du centre par Rouge3 ! Egalite ! Rouge3, but ! Attaque rapide de l'équipe rouge, Jaune10, La formation de l'équipe jaune est basée sur l'attaque par le centre. L'équipe japonaise a gagné dans le Groupe C du deuxième Tour, tandis que l'équipe allemande a gagné dans le Groupe D. Rouge1 prend la balle, mauvaise passe C'est l'équipe jaune qui relance le jeu, Magnifique dribble du Jaune2, Passe pour Jaune5. Est-ce que Jaune6 passe à Jaune5?

Figure 8: French output

covers a roughly 20 second period of a quater-final from RoboCup'97.

For comparison, we have included MIKE's French and Japanese descriptions of the same game period in Figure 8 and Figure 7. In general, the generated commentary differs because of the timing issues resulting from two factors: agent concurrency and the length of the NL-templates. One NL template is randomly chosen from several candidates at translation time and it is the length of this template that decides the timing of the next content selection.

5.2 Effect of Rearrangements

Importance Score Increase

Figure 9 plots the importance score of the propositions in MIKE's commentary for the some RoboCup'97 quater-final we used in the previous section. The horizontal axis indicates time unit of 100msec and the vertical axis the importance score of the comment being uttered (taking into account reductions due to interruption, abbreviation, or repeated use of a proposition). The solid line describes the importance score change with interruption, abbreviation and repetition, whereas the dotted shows that without such rearrangements. As we described in §4, the area between the plotted lines and the horizontal axis indicates the total importance score.

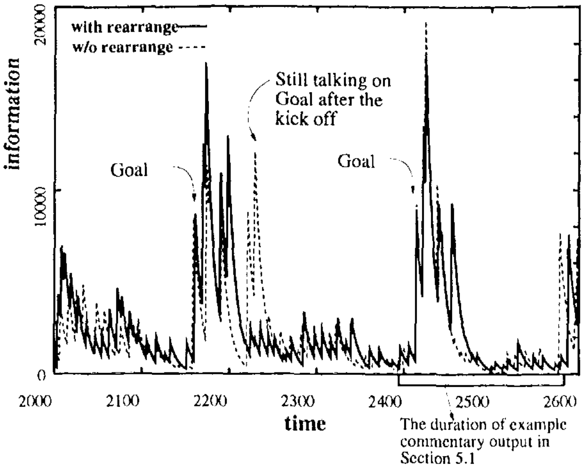

Figure 9: Importance score change during a RoboCup'97 quater-final game

Two observations:

- The graph peaks generally occur earlier for the solid line than for the dotted. This indicates that the commentary with rearrangements is more timely than the commentary that repeatedly selects the most important proposition. For instance, the peaks caused by a goal around time 2200 spread out for the dotted line, which is not the case for the solid line. Also, the peaks are higher for the solid line than dotted.
- The area covered by the solid line is larger than that by the dotted, meaning that the total importance score is greater with rearrangements. During this whole game, the total importance score with rearrangements amounted 9.90% more than that without.

Decrease of Delayed Utterances

As a further experiments, we manually annotated each statement in the Japanese output for the RoboCup'97 quater-final with it optimal time for utterance. We then calculated the average delay in the appearance of these statements in MIKE's commentary both with and without rearrangements. We found that adding the rearrangements decreased this delay from 2.51sec to 2.16sec , a improvement at about 14%.

6 Related Works

(Suzuki et al., 1997) have proposed new interaction styles to replace conventional goal-oriented dialogues. Their multi-agent dialogue system that chats with a human considers topics and goals as being situated within the context of interactions among participants. Their model of context handling is an adaptation of a subsumption architecture. One important common aspect between their system and MIKE is that the system itself creates topics.

The SOCCER system described in (Andrè et al., 1994), combines a vision system with an intelligent multimedia generation system to provide commentary on 'short sections of video recordings of real soccer games'. The system is built on VITRA, which uses generalized simultaneous scene description to produce concurrent image sequence evaluation and natural language processing. The vision system translates TV images into information and the intelligent multimedia generation module then takes this information and presents it by combining media such as text, graphics and video.

7 Conclusions and Future Work

We have described how MIKE, a live commentary generation system for the game of soccer, deals with the issues of real time content selection and realization.

MIKE uses heterogeneous modules to recognize various low-level and high-level features from basic input information on the positions of the ball and the players. An NL generator then selects contents from a large number of propositions describing these features.

The selection of contents is controlled by importance scores that intuitively capture the amount of information communicated to the audience. Under our principle of maximizing the total importance scores communicated to the audience, the decision on how a content should be realized considering rearrangements such as interruption, abbreviation, is decided at the same time as the selection of a content. Thus, one of our discoveries was that severe *when-to-say* restriction works to tightly incorporate *what-to-say* (content selection) module and a *how-to-say* (language realization) module.

We presented sample commentaries produced by MIKE in English, French and Japanese. The effect of using the rearrangements was shown compared and found to increase the total importance scores by 10%, to decrease delay of the commentary by 14%.

An important goal for future work is parameter learning to allow systematic improvement of MIKE's performance. Although the parameters used in the system should ideally be extracted from the game log corpus, this opportunity is currently very limited; only the game logs of RoboCup'97 (56 games) and JapanOpen-98 (26 games) is open to public. Additionally, no model commentary text corpus is available. One way to surmount the lack of appropriate corpora is to utilize feedback from an actual audience. Evaluations and requests raised by the audience could be automatically reflected in parameters such as the initial values for importance scores, rates of decay of these scores, the coefficients in the formulae used for controlling inferences.

Another important research topic is the incorporation of more sophisticated natural language generation technologies in MIKE to produce a more lively, diverse output. At the phrase generation level, this includes the generation of temporal expressions, anaphoric references to preceding parts of the commentary, embedded clauses. At the more surface level, these are many research issues related to text-to-speech technology, especially prosody control.

References

E. Andrè, G. Herzog, and T. Rist. 1994. Multimedia presentation of interpreted visual data. In P. McKevitt, editor, *Proceedings of AAAI-94, Workshop on Integration of Natural Language and Vision Processing*, pages 74–82, Seattle, WA.

D.E. Appelt. 1982. Planning natural-language referring expressions. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 108–112.

R.A. Brooks. 1991. A new approach to robotics. *Science*, pages 1227–1232.

DEC. 1994. Dectalk express text-to-speech synthesizer user guide.

Elan. 1997. Speech proverbe engine unit manual.

L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. 1980. The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2):213–253.

Fujitsu. 1995. FSUNvoice1.0 Japanese speech synthesizer document.

E.H. Hovy. 1988. *Generating Natural Language under Pragmatic Constraints.* Lawrence Erlbaum Associates.

I. Noda and H. Matsubara. 1996. Soccer Server and researches on multi-agent systems. In Hiroaki Kitano, editor, *Proceedings of IROS-96 Workshop on RoboCup*, pages 1–7, Nov.

N. Suzuki, S. Inoguchi, K. Ishii, and M. Okada. 1997. Chatting with interactive agent. In *Eurospeech'97*, volume 3, pages 2243–2247.

K. Tanaka-Ishii, I. Noda, I. Frank, H. Nakashima, K. Hasida, and H. Matsubara. 1998. Mike: An automatic commentary system for soccer. In *Proceedings of ICMAS98*, Paris, France.