# Generating multimedia presentations for RoboCup soccer games

**Elisabeth André, Gerd Herzog, Thomas Rist**

**Angaben zur Veröffentlichung / Publication details:**

André, Elisabeth, Gerd Herzog, and Thomas Rist. 1998. "Generating multimedia presentations for RoboCup soccer games." In *RoboCup-97: Robot Soccer World Cup I*, edited by Hiroaki Kitano, 200–215. Berlin [u.a.]: Springer. https://doi.org/10.1007/3-540-64473-3_61.



# Generating Multimedia Presentations for RoboCup Soccer Games

Elisabeth André, Gerd Herzog, and Thomas Rist

DFKI GmbH, German Research Center for Artificial Intelligence
D-66123 Saarbrücken, Germany
{andre,herzog,rist}@dfki.de

**Abstract.** The automated generation of multimedia reports for time-varying scenes on the basis of visual data constitutes a challenging research goal with a high potential for many interesting applications. In this paper, we report on our work towards an automatic commentator system for RoboCup, the Robot World-Cup Soccer. ROCCO (RoboCup-Commentator) is a prototype system that has emerged from our previous work on high-level scene analysis and intelligent multimedia generation. Based on a general conception for multimedia reporting systems, we describe the initial ROCCO version which is intended to generate TV-style live reports for matches of the simulator league.

## 1 Introduction

Intelligent robots able to form teams and play soccer games are the outspoken vision of the Robot World-Cup Soccer (RoboCup) program which has been initiated by Kitano [12] in order to revitalize and foster research in AI and robotics. From a research point of view the RoboCup program stands for a number of challenges such as agent and multiagent systems, real-time recognition, planning and reasoning, behavior modeling and learning, etc.

The realization of an automated reporting system for soccer games is the long-term vision of a research activity called SOCCER which started in 1986 at University of the Saarland as part of the VITRA (Visual Translator) project. The research challenges addressed in VITRA-SOCCER include the qualitative interpretation of a continuous flow of visual data, and the automated generation of a running report for the scene under consideration. Short sections of video recordings of soccer games have been chosen as a major domain of discourse since they offer interesting possibilities for the automatic interpretation of visual data in a restricted domain. Figure 1 shows a screen hardcopy of the first VITRA-SOCCER prototype which was implemented in 1988 [3]. In the early 90's the focus of interest moved towards the combination of techniques for scene interpretation [3, 7–9] and plan-based multimedia presentation design which has been addressed by DFKI [2, 5]. This was a crucial move in VITRA-SOCCER since it opened the door to an interesting new type of computer-based information system that provides highly flexible access to the visual world [4]. Vice versa, the

broad variety of commonly used presentation forms in sports reporting provides a fruitful inspiration when investigating methods for the automated generation of multimedia reports.

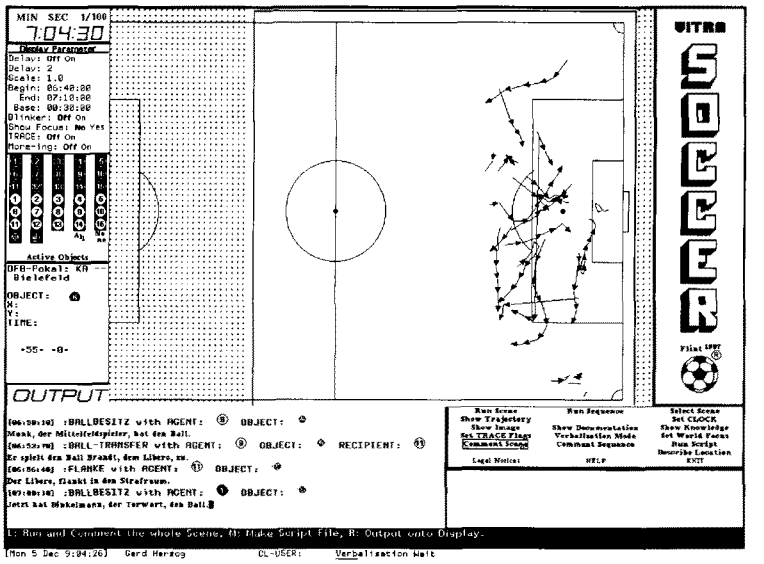


**Fig. 1.** The basic windows of the first VITRA-SOCCER system

Apparently, the combination of both visions brings into mind the automated reporter that presents and comments soccer games played by robots. But is it worth at all to spend effort on building such a reporting system? From the point of view of VITRA-SOCCER the answer is clearly yes, because RoboCup provides an excellent testbed for both, high-level interpretation of time-varying scenes and the automated generation of multimedia presentations for conveying recognized events.

The first part of this contribution describes the VITRA-SOCCER conception for transforming visual data into multimedia presentations such as TV-style live reports. This conception also forms the basis for the RoboCup commentator system ROCCO which will be presented in more detail in the subsequent sections.

## 2 From Visual Data to Multimedia Presentations

The automatic description of a time-varying scene by generating elaborated multimedia reports from sensory raw data constitutes a multistage process. In the following, we describe a decomposition of the transformation process into maintainable subtasks. A rough decomposition is depicted in Fig. 2 where the transformation process is subdivided into three distinct levels of processing. The figure

also provides a view on the heterogeneous representation formats as they will be used to bridge between the different steps of the transformation.

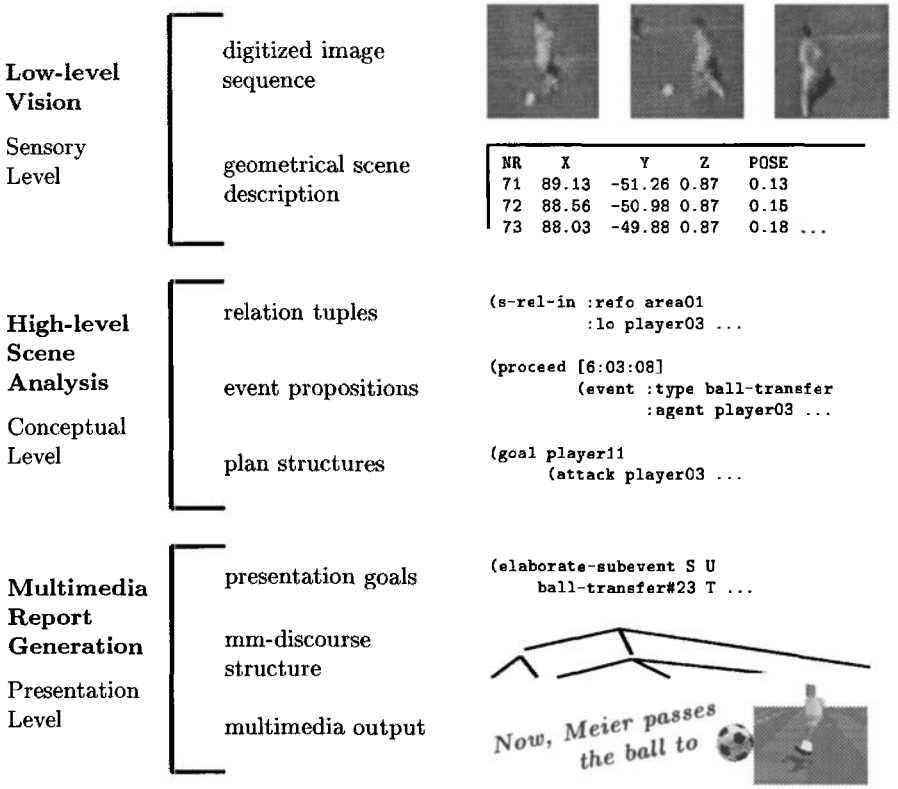


**Fig. 2.** Levels of representation

## 2.1 Low-level Vision

The processes on the sensory level start from digitized video frames (cf. Fig. 3) and serve for the automated construction of a symbolic computer-internal representation of the perceived scene.

This processing of image sequences—as it is to be carried out by a vision system—concentrates on the recognition and tracking of visible objects. For VITRA-SOCCER, sequences of up to 1000 images (40 seconds play back time) recorded with a stationary TV-camera during a game in the German professional soccer league, have been evaluated by our project partners at FhG-IITB in Karlsruhe [8]. The employed vision system performed a segmentation and cueing of moving objects by computing and analyzing displacement vector fields. The

calibration of the camera allowed for the transformation of trajectories from the image plane into world coordinates. In the soccer domain, segmentation becomes quite difficult because the moving objects cannot be regarded as rigid and occlusions occur very frequently. The as yet partial trajectories delivered by the vision component (see Fig. 1) required some manual completion in order to ensure a proper assignment of recognized object candidates to previously known players and the ball.

**Fig. 3.** Image sequence showing an attack in a soccer game

Within the initial version of the system all mobile objects have been represented as centroids since a full 3D-reconstruction of the players from the kind of image sequence shown in Fig. 3 is simply not possible yet (see also [10] for related work using American football scenes). However, the work in [15] describes research on the model-based 3D-reconstruction of the movements of articulated bodies. With this approach, a cylindric representation and a kinematic model of human walking, which is based on medical data, is utilized for the incremental recognition of pedestrians. As can be seen in Fig. 4, the algorithm determines the 3D positions as well as the postures of a single moving person within a real-world image sequence. In addition to treating this example domain in VITRA as well [7], the approach for the geometric modeling of an articulated body has also been adopted in order to represent the players in the soccer domain.

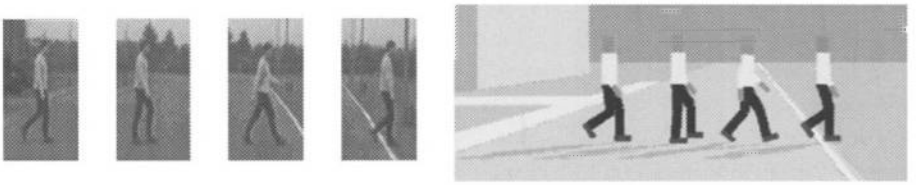

**Fig. 4.** Automatic recognition of a walking person

As an intermediate representation between low-level image analysis and high-level scene analysis, we rely on the concept of the so-called *geometrical scene description* (GSD) which has been introduced by Neumann [13] as a representation for the intended output of the low-level vision process. The aim of this description format is to represent the original image sequence completely and without loss of information, i.e. the data in the GSD suffice (in principle) to reconstruct the raw images. Basically, the GSD contains information concerning visible objects and their locations over time, together with additional world knowledge about the objects. The GSD constitutes an idealized interface for an intelligent reporting system which is supposed to sit on top of a vision system. In applications, like our VITRA system, the GSD is restricted according to the practical needs. In VITRA, for example, only the trajectories of the moving objects are provided by the vision component. The stationary part of the GSD, an instantiated model of the static background, is fed into the system manually.

### 2.2 High-Level Scene Analysis

High-level scene analysis provides an interpretation of visual information and aims at recognizing conceptual units at a higher level of abstraction. The information structures 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, representations of recognized object movements, and further higher-level concepts such as representations of behaviour and interaction patterns of the agents observed.

As indicated at the end of Sect. 2.1, the GSD provides the input on which VITRA-SOCCER performs high-level scene analysis. Within the GSD spatial information is encoded only implicitly. In analogy to prepositions, their linguistic counterparts, spatial relations provide a qualitative description of spatial aspects of object configurations [1]. Each spatial relation characterizes a class of object configurations by specifying conditions, such as the relative position of objects or the distance between them.

The characterization and interpretation of object movements in terms of motion events serves for the symbolic abstraction of spatio-temporal aspects of a time-varying scene. In VITRA-SOCCER the recognition of such motion events is based on generic event models, i.e., declarative descriptions of classes of interesting object movements and actions [3]. Besides the question of which events are to be extracted from the GSD, it is decisive how the recognition process is realized. The most distinguishing feature of the VITRA-SOCCER approach is that it relies on an *incremental* strategy for event recognition. Rather than assuming a complete GSD before starting the recognition process (e.g. [13]), VITRA-SOCCER assumes a GSD which will be constructed step by step and processed simultaneously as the scene progresses [8]. An example of an event definition and a sketch of the recognition process is provided in Sect. 4.

For human observers the interpretation of visual information also involves inferring the intentions, i.e. the plans and goals, of the observed agents (e.g., player A does not simply *approach* player B, but he *tackles* him). In the soccer domain

the influence of the agents' *assumed* intentions on the results of the scene analysis is particularly obvious. Given the position of players, their team membership and the distribution of roles in standard situations, stereotypical intentions can be inferred for each situation. Plan-recognition techniques may be employed for the automatic detection of presumed goals and plans of the acting agents [14]. With this approach, each element of the generic plan library contains information about necessary preconditions of the (abstract) action it represents as well as information about its intended effect. A hierarchical organization is achieved through the decomposition and specialization relation. Observable events and spatial relations constitute the leaves of the plan hierarchy.

### 2.3 Multimedia Report Generation

The presentation component of a reporting system can be regarded as a special instantiation of the class of intelligent multimedia presentation systems as defined in [6].

The interpretation of the perceived time-varying scene together with the original input data provides the required base material for the actual report generation. Depending on the user's information needs the system has to decide which propositions from the conceptual level should be communicated, to organize them and distribute them on several media. To accomplish these subtasks, operator-based approaches have become more and more popular in the User Interfaces community since they facilitate the handling of dependencies between choices (cf. [2, 5]).

The next step is the media-specific presentation of information. In the simplest case, presentation means automatic retrieval of already available output units, e.g, canned text or recorded video clips. More ambitious approaches address the generation from scratch. In VITRA-SOCCER, we were able to base the generation of visual presentations on the camera-recorded visual data and on information obtained from various levels of image interpretation. For example, when generating live reports, original camera data may be directly included in the presentation by forwarding them to a video window. To offer more interesting visualization techniques, we implemented algorithms for data aggregation, display style modifications, and the visualization of inferred information.

The task of natural-language (NL) output generation is usually divided into *text design* and *text realization*. Text design refers to the organization of input elements into clauses. This comprises the determination of the order in which the given input elements can be realized in the text and lexical choice. The results of text design are preverbal messages. These preverbal messages are the input for the *text realization* subtask which comprises grammatical encoding, linearization and inflection. Depending on the selected output mode, formulated sentences are then displayed in a text window or piped to a speech synthesis module. Several approaches to the generation of NL output have been tested within VITRA-SOCCER. Besides templates, we used a text realization component which was based on the formalism of Lexicalized LD/LP Tree Adjoining Grammars (TAG, [11]).

### 2.4 Task Coordination

An identification of subtasks as described above gives an idea of the processes that a multimedia reporting system has to maintain. The architectural organization of these processes is a crucial issue, especially when striving for a system that supports various presentation styles. For example, the automatic generation of live presentations calls (1) for an incremental strategy for the recognition of object movements and assumed intentions, and (2) for an adequate coordination of recognition and presentation processes. Also, there are various dependencies between choices in the presentation part. To cope with such dependencies, it seems unavoidable to interleave the processes for content determination, media selection and content realization.

## 3 RoCCo: A RoboCup Soccer Commentator System

A practical goal of our current activities is the development of a prototype system called ROCCO (RoboCup Commentator) that generates reports for RoboCup soccer games. As a first step, we will concentrate on matches of the simulator league which involves software agents only (as opposed to the different real robot leagues).

Technically, we rely on the RoboCup simulator called SOCCER SERVER [12], which is a network-based graphic simulation environment for multiple autonomous mobile robots in a two-dimensional space. The system provides a virtual soccer field and allows client programs (i.e. software robots) to connect to the server and control a specific player within the simulation (cf. Fig. 5). The SOCCER SERVER environment includes in particular a graphical user interface to monitor object movements and the control messages. In addition, independent components in support of a three-dimensional visualization will be supplied as well (cf. [12]).

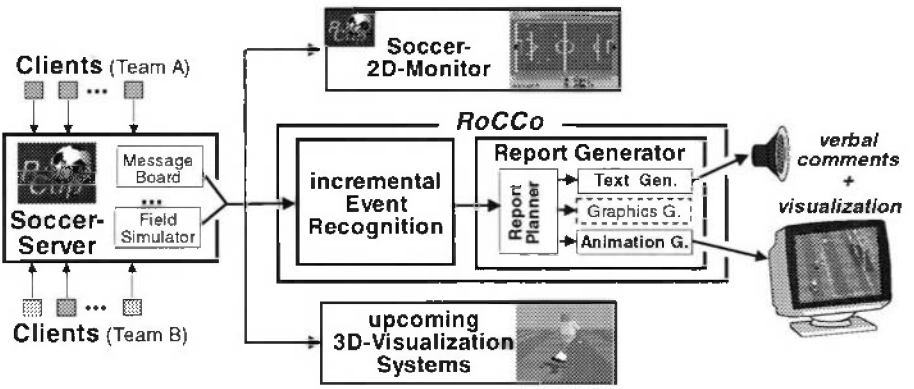


**Fig. 5.** Connecting SOCCER SERVER and ROCCO

The internal communication protocol used between simulator kernel and the monitor component can be exploited to access dynamic information concerning the time-varying scene generated through the simulation. These data serve as the geometrical scene description to be fed into our commentator system.

Depending on basic criteria like the information requirements (complete description vs. summary), the reporting mode (simultaneous vs. retrospective), and the available media for presentation, a broad spectrum of potential presentation styles exists for a system like ROCCO. From the variety of commonly used presentation forms in sports reporting we have selected TV-style live reports as a starting point. To enable this reporting style, ROCCO will comprise modules for the generation of text and 2D-animations (cf. Fig. 5). In its first version, ROCCO will generate a running report of the game and serve as an alternative for the original soccer monitor. For a further version we plan to incorporate a graphics generator in order to support additional reporting styles such as illustrated summaries.

## 4 Interpretation of RoboCup Soccer Scenes

The requirements for the event recogniton component of ROCCO are quite similar to those which have guided the design of VITRA-SOCCER. First of all, ROCCO should be able to produce *live reports*. That is the stream of continuously updated state information about an ongoing soccer game must be processed simultaneously as the game progresses. Moreover, if the presentation is to be focused on what is currently happening, it is very often necessary to describe object motions even while they occur. To enable such descriptions motion events have to be recognized stepwise as they progress and event instances must be made available for further processing from the moment they are noticed first. Therefore, VITRA-SOCCER's approach to the incremental recognition of events seems quite appropriate for the new task, too.

Furthermore, it is desirable to facilitate the definition and modification of event concepts. This is especially important because the set of interesting motion events in the simulation setting are certainly not completely identical to the motion events which we tried to capture from image sequences of real soccer games. Rather than coding event concepts in a pure procedural manner, VITRA-SOCCER allows for declarative descriptions of classes of higher conceptual units capturing the spatio-temporal aspects of object motions. The event concepts are organized into an abstraction hierarchy, grounded on specialization (e.g., *running* is a *moving*) and temporal decomposition. Of course, this kind of modeling event concepts seems also appropriate for ROCCO.

### 4.1 Definition of Events

Event concepts represent a priori knowledge about typical occurrences in a scene. To define event concepts we use representation constructs as shown in (cf. Fig. 6). The *header* slot indicates the event type to be defined (here *ball-transfer*),

and the number and types of objects which are involved in the occurrence of such an event. The slot *conditions* may be used to provide further constraints on variable instantiations. For example, according to the definition, a *ball-transfer* event involves the ball and two players which must belong to the same team. The slot *subconcepts* lists all involved sub-events. Roughly speaking one can say that the occurrence of the event implies occurrences of all its subconcepts. However, the reverse syllogism does not necessarily hold since the temporal relationships between the occurrences must be considered. For this purpose the event definition scheme comprises the slot *temporal-relations* which allows for the specification of timing information in an interval-based logics. For example, a *ball-transfer* event can only be recognized, if the three specified subevents occur sequentially without interuption.

**Header:** (ball-transfer ?p1*player ?b*ball ?p2*player)
**Conditions:** (eql (team ?p1) (team ?p2))
**Subconcepts:** (has-ball ?p1 ?b) [I1]
(move-free ?b) [I2]
(has-ball ?p2 ?b) [I3]
**Temporal-Relations:** [I1] :meets [ball-transfer]
[I1] :meets [I2]
[I2] :equal [ball-transfer]
[I2] :meets [I3]

**Fig. 6.** Event model for *ball-transfer*

The decompositional approach to event modeling results in a hierarchical organisation of event definitions. The lowest level of this hierarchy is formed by elementary events, such as *move*, that can be calculated directly from object positions which are recorded within the GSD.

### 4.2 Event Recognition

The recognition of an event occurring in a particular scene corresponds to an instantiation of the respective generic event concept. In order to enable an incremental recognition of occurrences the declarative definitions of event concepts are compiled into so-called *course diagrams*. Course diagrams are labeled directed graphs which are used to internally represent the prototypical progression of an event. The recognition of an occurrence can be thought of as traversing the course diagram, where the edge types are used for the definition of the basic event predicates.

Since the distinction between events that have and those that have not occurred is insufficient, we have introduced the additional predicates *start*, *proceed* and *stop* which are used for the formulation of traversing conditions in the course diagrams. In contrast to the event specifications described above, course

diagrams rely on a discrete model of time, which is induced by the underlying sequence of digitized video frames. Course diagrams allow incremental event recognition, since exactly one edge per unit of time is traversed. However, by means of constraint-based temporal reasoning, course diagrams are constructed automatically from the interval-based concept definitions [9].

The derived course diagram for the concept *ball-transfer* is shown in Fig. 7. As can be seen from the labels of the edges and the traversing conditions the event starts if a *has-ball* event stops and the ball is free. The event proceeds as long as the ball is moving free and stops when the recipient (who must be a teammate) has gained possession of the ball.

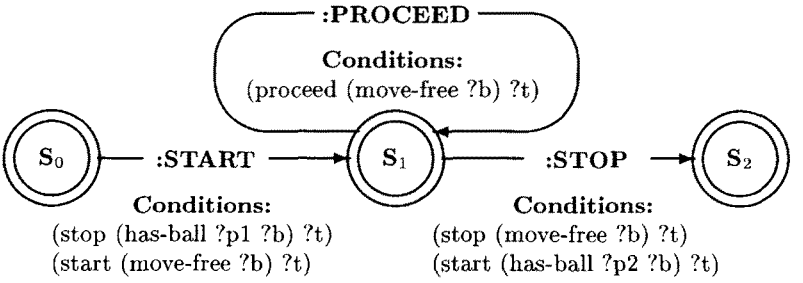


**Fig. 7.** Course diagram for `ball-transfer`

Course diagrams serve as recognition automata. As soon as new sensory data are provided, the recognition component continues with the traversal of already activated course diagrams and tries to trigger new ones. An activated course diagram corresponds to exactly one potential instantiation of a generic event. Consequently, several recognition automata for the same event type may run concurrently to capture simultaneous occurrences of a certain event type.

## 5 Report Planning

To plan the contents and the structure of a soccer report, we rely on the plan-based approach which has been originally developed for the WIP presentation system [2, 5]. The main idea behind this approach is to formalize action sequences for composing multimedia material as operators of a planning system. The effect of a planning operator refers to a complex communicative goal (e.g., describe the scene) while the expressions in the body specify which communicative acts have to be executed in order to achieve this goal.

Starting from a presentation goal, the planner looks for operators whose effect subsumes the goal. If such an operator is found, all expressions in the body of the operator will be set up as new subgoals. The planning process terminates if all subgoals have been expanded to elementary generation tasks which are

forwarded to the media-specific generators. For the first prototype of ROCCO the main emphasis will be on the planning of verbal comments which will be realized by a text generator.

To utilize the plan-based approach for RoboCup reports, we had to define additional strategies for scene description. For example, the strategy shown in Fig. 8 may be used to verbally describe a sequence of events by informing the user about the main events (e.g., *team-attack*) and to provide more details about the subevents (e.g., *kick*). While *s-inform* is an elementary act that is directly forwarded to the text generator, *elaborate-subevents* has to be further refined, e.g., by applying the strategy shown in Fig. 9. It informs the user about all salient subevents and provides more details about the agents involved. To determine the salience of an event, factors such as its frequency of occurrence, the complexity of its generic event model, the salience of involved objects and the area in which it takes place are taken into account (see also [3]). All events are described in their temporal order.

```
Header: (Describe-Scene S U ?events)
Effect: (FOREACH ?one-ev
             WITH (AND (BEL S (Main-Ev ?one-ev))
                       (BEL S (In ?one-ev ?events)))
             (BMB S U (In ?one-ev ?events)))
Applicability-Conditions:
        (BEL S (Temporally-Ordered-Sequence ?events))
Inferior Acts:
        ((FOREACH ?one-ev
             WITH (AND (BEL S (Main-Ev ?one-ev))
                       (BEL S (In ?one-ev ?events)))
             (S-Inform S U ?one-ev)
             (Elaborate-Subevents S U ?one-ev)))
```

**Fig. 8.** Plan operator for describing a scene

The strategies defined in Fig. 8 and Fig. 9 can be used to generate a posteriori scene descriptions. They presuppose that the input data from which relevant information has to be selected are given a priori. Since both strategies iterate over *complete* lists of temporally ordered events, the presentation process cannot start before the interpretation of the whole scene is completed.

However, ROCCO should also be able to generate live reports. Here, input data are continuously delivered by a scene interpretation system and the presentation planner has to react immediately to incoming data. In such a situation, no global organization of the presentation is possible. Instead of collecting scene data and organizing them (e.g., according to their temporal order as in the first two strategies), the system has to locally decide which event should be reported next considering the current situation. Such behavior is reflected by the strategy shown in Fig. 10. In contrast to the strategy shown in Fig. 9, events are selected

```
Header: (Elaborate-Subevent S U ?ev)
Effect: (FOREACH ?sub-ev
             WITH (AND (BEL S (Salient ?sub-ev))
                       (BEL S (Sub-Ev ?sub-ev ?ev)))
             (BMB S U (Sub-Ev ?sub-ev ?ev)))
Applicability-Conditions:
        (AND (BEL S (Sub-Events ?ev ?sub-events))
             (BEL S (Temporally-Ordered-Sequence ?sub-events)))
Inferior Acts:
        ((FOREACH ?sub-ev
             WITH (AND (BEL S (In ?sub-ev ?sub-events))
                       (BEL S (Salient ?sub-ev)))
             (S-Inform S U ?sub-ev)
             (Elaborate-Agents S U ?sub-ev)))
```

**Fig. 9.** Plan operator for describing subevents

for their topicality. Topicality is determined by the salience of an event and the time that has passed since its occurrence. Consequently, the topicality of events decreases as the scene progresses. If an outstanding event (e.g., a goal kick) occurs which has to be verbalized as soon as possible, the presentation planner may even give up partially planned presentation parts to communicate the new event as soon as possible.

```
Header: (Describe-Next S U ?ev)
Effect: (AND (BMB S U (Next ?preceding-ev ?ev))
             (BMB S U (Last-Reported ?ev)))
Applicability-Conditions:
        (AND (BEL S (Last-Reported ?preceding-ev))
             (BEL S (Topical ?ev *Time-Available*))
             (BEL S (Next ?preceding-ev ?ev)))
Inferior Acts:
        ((S-Inform S U ?ev)
         (Describe-Next S U ?next-ev))
```

**Fig. 10.** Plan operator for simultaneous description

## 6 Generation of Verbal Comments

VITRA-SOCCER was only able to generate complete, grammatically well-formed sentences. As a consequence, the output was not tuned to live reports that often contain short, sometimes syntactically incorrect phrases. Unlike VITRA-SOCCER, ROCCO aims at the generation of a large variety of natural-language

expressions that are typical of the soccer domain. Since it is rather tedious to specify soccer slang expressions in formalisms like TAG, we decided to use a template-based generator instead of fully-fledged natural-language design and realization components. 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 soccer TV reports in English have been transcribed and annotated. Each event proposition is assigned a list of templates that may be used to verbalize them. For instance, Table 1 lists some templates that may be used to verbalize an instance of a finished *ball-transfer* event in a live report.

| Template | Constraints | Vb | Floridity | Sp | Formality | Bias |
|---|---|---|---|---|---|---|
| (?x passes the ball to ?y) | None | 8 | dry | 4 | formal | neutral |
| (?x playes the ball towards ?y) | None | 8 | dry | 4 | formal | neutral |
| (?x towards ?y) | (NotTop ?x) | 5 | dry | 3 | slang | neutral |
| (?x combines with ?y) | None | 6 | normal | 3 | slang | neutral |
| (?x now ?y) | (NotTop ?x) | 5 | dry | 2 | slang | neutral |
| (ball played towards ?y) | None | 5 | dry | 3 | colloquial | neutral |
| (the ball came from ?x) | (NotTop ?x) | 6 | dry | 3 | colloquial | neutral |
| (a touch by ?x) | None | 5 | normal | 2 | slang | neutral |
| (turn from ?x) | None | 4 | dry | 2 | slang | neutral |
| (shot) | None | 1 | dry | 1 | colloquial | neutral |
| (?x) | (NotTop ?x) | 2 | dry | 1 | slang | neutral |
| (now ?y) | (NotTop ?y) | 3 | dry | 1 | slang | neutral |
| (?y was there) | None | 4 | dry | 1 | slang | neutral |
| (?y) | (NotTop ?y) | 2 | dry | 1 | slang | neutral |
| (well done) | None | 2 | dry | 0 | colloquial | positive |
| (good work from ?x) | None | 5 | normal | 1 | colloquial | positive |
| (a lovely ball) | None | 3 | flowery | 1 | colloquial | positive |

**Table 1.** Some templates for the event *ball-transfer*

Constraints specify whether a certain template can be employed in the current context. For instance, *"Now Miller"* should not be uttered if Miller is already topicalized. Currently, we assume that the whole field is visible. Thus we do not get any constraints referring to the visual focus. To select among several applicable templates, the following features are considered:

**Verbosity:** The verbosity of a template depends on the number of words it contains. While instantiated slots correspond to exactly one word, non-instantiated slots have to be forwarded to the nominal phrase generator which decides what form of nominal phrase is most appropriate. Since the length is not yet known at that point in time, ROCCO assumes a default word number of two for non-instantiated slots.

**Floridity:** We distinguish between dry, normal and flowery language. Flowery language is composed of unusual ad hoc coinages, such as *"a lovely ball"*. Templates marked as normal may contain metaphors, such as *(finds the gap)*, while templates marked as dry, such as *(playes the ball towards ?y)* just convey the plain facts.

**Specifity:** The specifity of a template depends on the number of verbalized deep cases and the specifity of the natural-language expression chosen for the action type. For example, the specifity of *(?x looses the ball to ?y)* is 4 since 3 deep cases are verbalized and the specifity of the natural-language expression referring to the action type is 1. The specifity of *(misdone)* is 0 since none of the deep cases occurs in the template and the action type is not further specified.

**Formality:** This attribute can take on the values: slang, colloquial and normal. Templates marked as formal are grammatically correct sentences which are more common in newspaper reports. Colloquial templates, such as *"ball played towards Meier"*, are simple phrases characteristic of informal conversation. Slang templates are colloquial templates peculiar to the soccer domain, such as *"Miller squeezes it through"*.

**Bias:** Biased templates, such as *(well done)* contain an evaluation of an action or event. Bias may be positive, negative or neutral.

While the features listed above don't change within a game and can be computed before generating a report, *dynamic features*, such as *How often mentioned?* and *Last mentioned?*, have to be continuously updated during the game.

To select a template, Rocco first determines all templates associated with an event concept, checks for which of them the constraints are satisfied and subsequently performs a four-phase filtering process. Only the best templates of each filtering phase will be considered for the next evaluation step.

1. *Compute the effectivity rate of all applicable templates:*
   Set the effectivity rate of all templates to 0 and punish or reward them by increasing or decreasing their effectivity rate considering their length and specifity. If the system is under time pressure, a template will be punished for its length, but rewarded for its specifity. In phases with little activity, all templates will get a reward both for their length and specifity. In all other situations, only the specifity of a template will be rewarded.
2. *Compute the variability rate of the most effective templates:*
   Punish templates that have been recently or frequently been used. Unusal templates, i.e., templates that seldom occur in a soccer report, get an additional punishment.
3. *Convey the speaker's partiality:*
   Select templates which convey the speakers partiality best. That is if the system is in favor of team X, prefer positive templates for describing activities of X to neutral templates, and neutral templates to negative templates.
4. *Select templates that convey the style of presentation best:*
   For example, prefer slang templates to colloquial templates and colloquial templates to normal templates in case of a live report.

For illustration, let's suppose that ROCCO is not in favor of a particular team, under time pressure and has to verbalize the following event proposition:

*(ID123 :Type ball-transfer :Agent sp1 :object Ball :Recipient sp2 :State Finished :Start 5 :Finish 10)*

Let's further assume that *sp1* is topicalized. ROCCO first selects all applicable templates from the table shown as Table 1. *(?x towards ?y)*, *(the ball came from ?x)*, *(?x now ?y)*, and *(?x)* are not applicable because *sp1* is topicalized. The remaining templates are checked for their effectivity. Only four templates pass this test: *(ball played towards ?y)*, *(shot)*, *(now ?y)* and *(?y)*. For the purpose of this example, we assume that the system was under permanent time pressure and that *(shot)*, *(now ?y)* and *(?y)* have already been used very often. Therefore, after the variability test only the candidate *(ball played towards ?y)* is left, and there is no need to apply further filters. The remaining template is taken for verbalization and the value for the non-instantiated slot *?y* is forwarded to the nominal phrase generator that takes into account the discourse context to decide whether to generate a pronoun or a noun phrase (with modifiers).

## 7 Conclusion

In this contribution, we have reported on our efforts towards the development of ROCCO, a system that observes RoboCup soccer simulations in order to provide informative multimedia reports on recognized occurrences. The conception of ROCCO builds on the experience gained from our long-term research in the context of VITRA-SOCCER. The new system ROCCO follows an incremental strategy for the recognition of higher-level concepts, such as spatial relations and motion events, and relies on a plan-based approach to communicate recognized occurrences with multiple presentation media (currently written text, spoken language, and 2D-visualizations).

Given our specific interest in high-level scene analysis and automated multimedia report generation, we will benefit in particular from the RoboCup initiative. The intrinsic difficulty of low-level image analysis leads to limited availability of suitable data material which has always been hindering our experimental investigations in the context of VITRA-SOCCER. RoboCup promises to ameliorate this situation. The broad use of a common simulation environment will certainly contribute to a much better research infrastructure which we can use as a flexible testbed for our current and future work on the automated generation of multimedia reports for time-varying scenes.

## Acknowledgements

We would like to thank Hiroaki Kitano for encouraging us to start the ROCCO project. Furthermore, we are grateful to Dirk Voelz for transcribing and annotating the soccer TV live reports and his work on the implementation of the ROCCO system.

## References


1. E. André, G. Bosch, G. Herzog, and T. Rist. Coping with the Intrinsic and the Deictic Uses of Spatial Prepositions. In K. Jorrand and L. Sgurev, editors, *Artificial Intelligence II: Methodology, Systems, Applications*, pages 375–382. North-Holland, Amsterdam, 1987.
2. E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster. WIP: The Automatic Synthesis of Multimodal Presentations. In M. T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 75–93. AAAI Press, Menlo Park, CA, 1993.
3. E. André, G. Herzog, and T. Rist. On the Simultaneous Interpretation of Real World Image Sequences and their Natural Language Description: The System SOCCER. In *Proc. of the 8th ECAI*, pages 449–454, Munich, Germany, 1988.
4. E. André, G. Herzog, and T. Rist. Multimedia Presentation of Interpreted Visual Data. In P. Mc Kevitt, editor, *Proc. of AAAI-94 Workshop on "Integration of Natural Language and Vision Processing"*, pages 74–82, Seattle, WA, 1994. Also available as Report no. 103, SFB 314 – Project VITRA, Universität des Saarlandes, Saarbrücken, Germany.
5. E. André and T. Rist. Towards a Plan-Based Synthesis of Illustrated Documents. In *Proc. of the 9th ECAI*, pages 25–30, Stockholm, Sweden, 1990.
6. M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A Standard Reference Model for Intelligent Multimedia Presentation Systems. *Computer Standards & Interfaces*, 1998. To appear in the Special Issue on Intelligent Multimedia Presentation Systems.
7. G. Herzog and K. Rohr. Integrating Vision and Language: Towards Automatic Description of Human Movements. In I. Wachsmuth, C.-R. Rollinger, and W. Brauer, editors, *KI-95: Advances in Artificial Intelligence. 19th Annual German Conference on Artificial Intelligence*, pages 257–268. Springer, Berlin, Heidelberg, 1995.
8. G. Herzog, C.-K. Sung, E. André, W. Enkelmann, H.-H. Nagel, T. Rist, W. Wahlster, and G. Zimmermann. Incremental Natural Language Description of Dynamic Imagery. In C. Freksa and W. Brauer, editors, *Wissensbasierte Systeme. 3. Int. GI-Kongreß*, pages 153–162. Springer, Berlin, Heidelberg, 1989.
9. G. Herzog and P. Wazinski. VIsual TRAnslator: Linking Perceptions and Natural Language Descriptions. *AI Review*, 8(2/3):175–187, 1994.
10. S. Intille and A. Bobick. Visual Tracking Using Closed-Worlds. In *Proc. of the 5th Int. Conf. on Computer Vision*, pages 672–678, Cambridge, MA, 1995.
11. A. Kilger. Using UTAGs for Incremental and Parallel Generation. *Computational Intelligence*, 10(4):591–603, 1994.
12. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup: A Challenge Problem for AI. *AI Magazine*, 18(1):73–85, 1997.
13. B. Neumann. Natural Language Description of Time-Varying Scenes. In D. L. Waltz, editor, *Semantic Structures: Advances in Natural Language Processing*, pages 167–207. Lawrence Erlbaum, Hillsdale, NJ, 1989.
14. G. Retz-Schmidt. Recognizing Intentions, Interactions, and Causes of Plan Failures. *User Modeling and User-Adapted Interaction*, 1:173–202, 1991.
15. K. Rohr. Towards Model-based Recognition of Human Movements in Image Sequences. *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, 59(1):94–115, 1994.