CONTENT-BASED RECKONING FOR INTERNET GAMES

Jörg R. J. Schirra
Department of Computer Science
Otto von Guericke University
Universitätsplatz 2
D 39 106, Magdeburg
Germany
E-mail: joerg@isg.cs.uni-magdeburg.de

KEYWORDS
Dead reckoning, behaviour encoding & decoding,

ABSTRACT

Event concepts, which structure our understanding of agents' behaviours as well as and our verbal descriptions, can reduce the amount of messages in network games if they are employed in a content-based extension of dead reckoning for anticipating and communicating the game states among clients. An example case is described by adapting to the new task the necessary recognition and reconstruction routines for such "semantic" event concepts from prior work in an AI project dealing with the linking of natural language generation and computer vision. Beside reducing communication, analysing the players in terms of high-level concepts also opens a preview of dealing with dissociations on a more general level.

NETWORK GAMES AND DISSOCIATION

Messages can pass through the internet only with a certain speed leading often to a significant latency between the time of sending and the time of receiving. This is particularly disturbing in settings with a high amount of interaction as in computer games, when players all around the earth may try enjoying a common game with each others, and long distances are coupled with temporally highly demanding exchanges. If, for example, the updating of the players' positions in a shooter game is delayed, there is a severe danger that the game dissociates, i.e., that the actions of the players no longer belong to one unique context of interaction. Therefore, handling dissociation is one of the crucial technical questions when building internet-based games. So far, several methods have been proposed to deal with the danger of dissociation. Reducing the amount of information that has to be sent through the net is a basis for most of them, as the reduction of necessary bandwidth also lessens the temporal pressure on the communication channel.

Often, a multitude of local game contexts are used that are essentially autonomous and may deviate from each other to a certain degree. Only major differences need explicit synchronization. One of the most successful methods in this framework is Dead Reckoning (or more generally: "predictive contracts", Mellon and West 1995): essentially, the movements of all player characters are locally extrapolated by every participant. Deviations between the predictions of the movements of the "local character" and the "real" positions determined by the player trigger corresponding messages to the others – but only if they exceed a certain threshold. The receivers react by smooth correction movements of the corresponding characters in their respective game contexts. So far, the threshold of allowed deviation depends merely on geometric information, and it is always the same no matter what kind of behaviour in which context is considered.

CONTENT-BASED RECKONING

This paper examines the idea of using descriptions as in natural language for anticipating and communicating the game states between clients instead of geometric information alone. Our verbal descriptions of actions in games are usually structured in events that belong mainly to an intermediate level of complexity: they are richer than pure geometry, and less global than strategic notions. Therefore, they cover relatively large segments of time. They also define context-sensitive thresholds for tolerable deviations only dependent on relevance.

While the usual dead reckoning is easily performed on the description of the game states as already given – basically the coordinates and velocity vectors of the characters – dead reckoning based on high-level descriptions depends crucially on routines for encoding and decoding the behaviour of game characters into concepts underlying our own descriptions of what is going on: e.g., "running along some channel toward a trap door" in a role playing game, or "playing a double-pass with a team-mate in front of the opposing team's penalty area" in an online soccer game. Such routines are described in the next section, based on research in a classical sub-field of AI: natural language systems.

Predictive contracts allow local game clients to act rather independently from each other by calculating the players' probable behaviours, instead of using their real input. Correspondingly, in our approach "semantic" concepts of actions are decoded into concrete spatio-temporal instances of that behaviour, predicting the players' activities. At the same time, every game client analyses the actual behaviour of "its" player by encoding his/her concrete movements into the high-level concepts, and compares them with its own predictions. The *relevant* distinctions are determined indicating necessary corrections of the simulations. This revised architecture is introduced in the second section.

The paper ends with a sketch of the approach's open questions and its potential to stimulate future research.

Behaviour Recognition and Reconstruction in VITRA

To define in an operational manner what we mean in this context by "content" is the first step toward content-based reckoning. Essentially, we need procedures for encoding behaviour given by means of coordinates and velocity vectors into relevant concepts, and also for decoding them back. Fortunately, an existing implementation of the operational semantics of spatio-temporal verbs and prepositions from a natural language system is close enough to the multiplayer game setting that it can immediately be adapted for a soccer game without many changes. In the project VITRA (VIsual TRAnslator), we have studied the connection of natural language generation and computer vision in the scenario of a radio sports report (Herzog and Wazinski 1994). The domain is soccer games. Aiming at a continuous processing from video input to a fluid report in German, many approaches of low and high level computer vision had to be coordinated in VITRA with components for utterance planning, syntax generation, and pragmatic anticipations of the listeners' understanding. Since a "life report" setting was chosen, temporal restrictions also played an important role.

After calculating from the video signal the 3D-positions, forms, and types of "objects" (players and ball), an idealized representation is sufficient for the higher levels of behaviour recognition: the centres of gravity of the players in the 2D-plane of the soccer field in a bird's eye view (Schirra et al. 1987). Encoding the game states is based on the relative positions of the players and the ball with respect to each others and to the geometric and functional parts of the soccer field. Such static spatial relations, which can be articulated by means of locative prepositions like "in" or "to the right of", can efficiently be detected by mathematical applicability functions based on simple geometric concepts and part-whole relations. Applying such a function onto an idealized geometric game state (as given by object recognition) leads to a fuzzy applicability value in [0.0 .. 1.0] (Schirra 1993). In Figure 1, such a function is graphically represented (unmarked position ≡ applicability value 0.0; black ≡ 1.0).

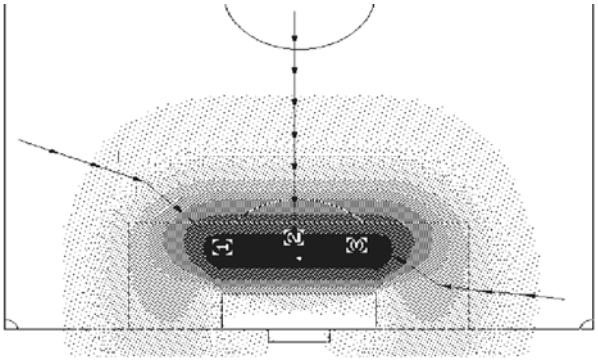

Figure 1: 2D-Typicality Distribution for "in front of" and Three Approximation Paths

More complex events (including simple and interactive behaviours as well as intentional acts) have been defined as temporal sequences of sets of such static spatial relations (plus part-whole relations like team membership), forming Finite State Machines (FSM), the states of which correspond to points in time, the transitions to the flow of time. They can immediately be used for parsing the input data into corresponding events. More precisely, we employed an augmented version of FSM: each state transition is marked twice (cf. Fig. 2): (a) with a condition, i.e., a conjunction of static spatial relations (or sub-events, see below) that have to hold at the time this transition is active, and (b) one of the following types: *start*, *proceed*, *succeed*, *stop.* The later classes determine phases of recognition: the moment of first assumption; an ongoing recognition without final confirmation (i.e., event not completed, completion can still fail); an ongoing confirmed recognition of *durative* events (e.g. "running"); and the final moment of successful recognition (first moment after the event). In order to simplify definitions, these recognition phases can be used for referring to sub-events in state transitions, too (in Fig. 2, the *start* phase of the "pass" event refers to the *stop* phase of an event "ball-possession", i.e., one player is losing contact with the ball).

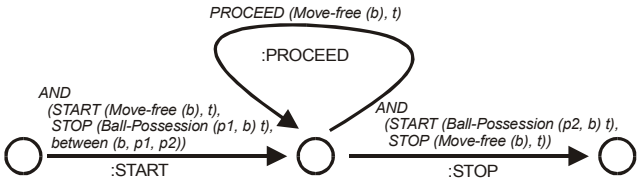

Figure 2: Augmented Finite State Machine for event type "player *p1* passes ball *b* to player *p2*"

The actual recognition of geometric events (e.g. "ball rolling toward something/somebody") or intentional acts (e.g. "scoring" or "attacking") is controlled in an object-oriented manner by means of "type demons" associated to each event type defined: if the conditions of its *start* phase apply to a sufficient degree (≥ 0.8) in the input of that moment, an event instance is created that tries to reach its *stop* phase through a couple of *proceed* (and perhaps *succeed*) loops. At every time step, a transition with fulfilled conditions must be made, or the recognition fails. While the instance is active, an event of that type is recognized as being *seemingly* happening – an assumption that might fail if the FSM does not reach a *succeed* or *stop* phase. Nevertheless, the assumption can already be communicated – covering a relatively long look-ahead on the probable development of the game in the future. The utterance has to be explicitly corrected only if the event recognition fails (Herzog 1992).

Most aspects of VITRA's language generation are not in the focus of attention here; it may suffice to mention that all event instances that are still active or successfully completed at that time are taken into consideration for generating the next utterance. They are dynamically ordered into a speech plan determined by criteria of the event types' relevance and the instances' topicality. The event concepts are linked in the lexicon to deep-case frames for verbs: the verbalization is finally created around that core (André et al. 1988).

Although it may on first view seem unexpected in the context of VITRA, we have also demonstrated that the very same data structures used for encoding – FSM and applicability functions – can efficiently be used for *de*coding corresponding verbal descriptions, i.e., for reconstructing the geometric scene. This problem was investigated in the context of listener modelling: How will a listener understand the utterance under planning? Such anticipations are useful in order to deal with several problems of linguistic pragmatics in language generation. In a nutshell: A corresponding "mental image" is constructed as the listeners' presumed

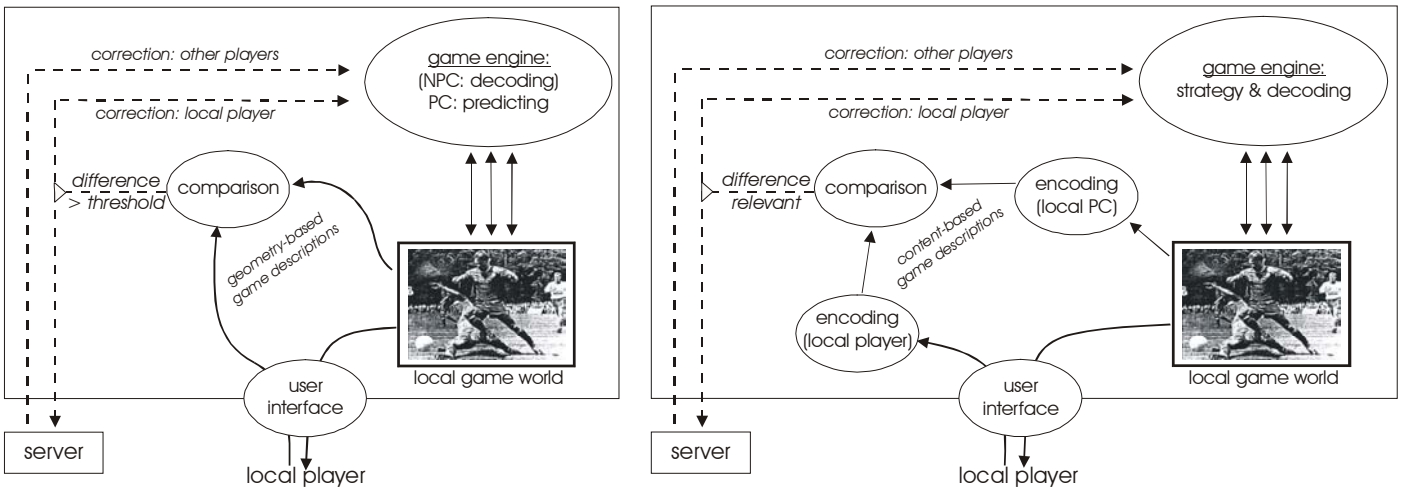

Fig. 3: Schematic Comparison of Standard Dead Reckoning (left) and Content-Based Reckoning (right)

understanding and compared to what actually happened on the soccer field; the differences are calculated and used to change the sentence finally uttered (Schirra 1995).

While encoding reduces the amount of irrelevant information, the problem with decoding is quite obviously that additional information has to be "invented" in some way. Here, the applicability (of a description for a given scene) has to be re-interpreted as a measure of typicality (of a scene for a given description): we are always looking for the most typical scene. Then, all objects mentioned are positioned so that all relations considered are maximally applicable. In order to reach that maximally typical scene from a given set of concepts, first the *proceed* and *succeed* transition loops of the corresponding FSMs – additionally marked for that purpose by temporal typicality distributions – are "expanded" in the current situational context, leading to a temporal sequence of sets of static spatial relations that have to hold simultaneously at one moment. Then, each of the sets can be transformed into adequate geometric information by means of a simple hill-climbing algorithm working on the applicability/typicality functions for the spatial relations. Like rubber bands, the typicality distributions "pull" the objects at highly typical positions: three such paths from different starting points are given in Fig. 1, indicating that the hill-climbing algorithm on spatial typicality distributions, which has also been successfully used in the author's group to plan camera paths in computer games (Halper et al. 2001), is highly context-sensitive. It is this context sensitivity that binds together the momentary pictures into the whole of the animation sequence: the result for one moment serves as the starting point for the next step. In this "cinematographic procedure", the typical movements of the objects involved in the context given are geometrically reconstructed in VITRA's listener model (Schirra and Stopp 1993).

Application to Network Games

While the routines in VITRA for encoding and decoding a soccer game in natural language were essentially developed in order to understand the cognitive foundations of speaking about something seen, most of its design principles can easily be adopted for a network computer game. Each player of that game may act as a member of a virtual soccer team. The player's essential activity is controlling the position (or velocity) of the character. He can also "perform" a few special activities, like kicking the ball. The input therefore consists in a constant stream of position/direction updates, punctured by a kicking action once and again. With respect to content-based reckoning, the soccer domain is intended as an example setting only.

Let us assume a server-client configuration. In the idealized standard dead reckoning algorithm (Fig. 3, left), the local game world is completely built by the client's game engine: the player characters (PC) are mainly controlled by extrapolating their prior movements – sometimes messages from the server initiate corrections. For non-player characters (NPC), we may assume the evaluation of behaviour scripts in the form of extended FSMs – similar to VITRA's decoding sequence described in the previous section. FSMs are indeed a widely used mechanism for determining the behaviour of NPCs in computer games. In this context, too, an idealization of the characters as points is basically sufficient: control points for the subsequent 3D-animation of the character on the screen have to be determined. Note that only in the NPCs' case strategic rules (e.g., from a SOAR-like component; Van Lent and Laird 1999) are employed by the game engine for selecting one of several event types applicable at that time.

The server receives a message from a client if the geometric difference between the local player's actions (input stream) and the predictions of that client's game engine exceeds a fixed threshold. The server forwards each such message with the correcting input coordinates to all the other clients triggering there the corresponding routines of adaptation. (Additional messages may indicate that no message was lost when corrections are superfluous for some time.)

Content-based reckoning (Fig. 3, right) uses the very same architecture extended by two modules for encoding geometry data into higher behavioural concepts: One of these modules "observes" the actual behaviour of the player (i.e., the input), the other one analyses the local game world encoding the PC's behaviour as generated by the game engine. In consequence, the comparison algorithm has to be changed, because now it is not geometric data to be compared with respect to a numeric threshold but complex actions, recognized ones as well as assumed ones, that either are the same in relevant aspects or not. We come back to the criteria of comparison in more detail soon.

Instead of numeric positions and velocity values, the correction messages now carry the information of actions taking place as in verbal descriptions. Of course, in the game setting, the event concepts do not have to be really verbalized in any natural language with all its peculiarities and redundancies. The event type's name, a mark for the current phase of the event recognition, some optional spatial relations indicating important location and direction parameters of that event instance, and a time stamp suffice for a message.

The game engine is modified in order to understand these messages. In general, the PCs' positions are now derived by means of decoding event concepts as described in the previous section, i.e., just like the NPCs' behaviours. On an intermediate level between the strategic overview and the concrete animation of the character, the gradient-based search for maximal typicality concretises behavioural concepts chosen by strategic rules in the current context delivering control parameters for the 3D-animations to be presented on the screen. However, this autonomous selection of actions may be overwritten by the concepts in correction messages – bringing in the *real* actions of the players. The correction itself is anchored in the current game state by the very same decoding procedures that unfold any behaviour of characters. Note that the geometric difference to be corrected cannot be extreme (or it would have been corrected earlier). The context-sensitivity of the cinematographic procedure therefore leads to smooth transitions even if the underlying concept is changed.

It is crucial here that events can be used in correction messages even if they are not yet completed, i.e., if they are merely assumed: events, the beginnings of which have been recognized in the players' movements (but not in the corresponding character's behaviour). Assume that the *start* phase of an event – e.g., of type "scoring", i.e. ball starts moving away from the player toward the opposing goal – was found in the player's activities but not in the local game with the PC's reckoned behaviour. Consequently, a correction message informs the game engines about this event, which then replaces the one currently active for that PC.

Time stamps and phase markers in the messages allow the game engines to position the correct parts of the expanded events at the right temporal frames. A combination with "time warp" techniques looks promising in order to simplify the integration of the events at the right time. Following (Mauve 2000), a network soccer game with its limited number of characters is a plausible application for a "time warp" addition to standard dead reckoning. It may work well with content-based reckoning, too. For virtual environments with more characters, however, "time warping" becomes too expensive.

As only the *start* phase has really happened so far in our example, most of the activity covered by that concept still belongs to the future at communication time. If the "scoring" action is finished as predicted (and communicated to the other clients) no further message needs to be sent for the rest of that time. Thus, the general frequency of messages is reduced as we have intended.

The crucial component is the comparison, filtering out relevant from irrelevant differences of behaviour. In principle we have to decide whether the two descriptions of behaviour constructed by the two encoding components are literally the same or not. In fact, in the original sports report setting, a similar comparison was performed in order to determine 13 kinds of difference between the listeners' anticipated understanding (the re-analysed mental image) and the real events (as seen by the reporter; Blocher and Schirra 1995). For content-based reckoning, only two cases have to be distinguished: (a) no instance of the event type covering the real behaviour is predicted, (b) two corresponding instances of the same event type occur, though with different parameters (mainly, a place or direction slot is filled with different spatial relations or the phase is wrong). In both cases the observed event must be communicated to the game engines. In the second case, a marker has to be added indicating that a corresponding instance with divergent parameters is already active and has to be adapted, not replaced. A third case seems interesting, too: no instance of a predicted event type is recognized in the real behaviour (case (a) inverted). However, this case is not relevant since it always co-occurs with a difference of type (a): then, as a consequence of the correction message, all active events of that PC are replaced.

The player's input may indeed differ to a greater or smaller degree from the positions generated by decoding the active concept: Only if the input cannot be "parsed" any longer into the same concept, a correction message is generated. Thus, depending on the concepts and the phases of the events, the tolerance allowed can be quite different – a few small steps only (e.g., if a lot of spatial relations restrict simultaneously the PC's position), or several long paces (if few relations are given or the PC is away from anything that could act as a reference object at that moment).

Some Problems

So far, an example architecture for content-based reckoning has been presented on the basis of procedures adapted from a project in cognitive science for recognizing and reconstructing events (i.e., behaviours in the game). Of course, a lot of questions remain open. For example: more complex event concepts lead not only to the positive effect of longer look-ahead times but also to higher chances of wrong recognition and more effort for recognizing. Therefore, determining for different game genres the right tactical level of detail and abstraction of the concepts is a particularly crucial task for further empirical research in this framework.

Content-based reckoning trades off less communication load for more local effort for encoding/decoding. These routines, however, become easily quite complicated – a standard problem in AI. While VITRA's routines for encoding and decoding at least performed approximately "in real time", other types of games with more characters or/and more difficult behaviours may become problematic. If the additional components could be transformed into "anytime" versions (Zilberstein 1996), thus leading even under temporal pressure to at least acceptable results, a dynamic adaptation to the resources available at *that* time (and for *that* client) offers an interesting solution. Typicality approximation with hill climbing has already "anytime" property. Encoding can also be transformed – though not easily (Wahlster et al. 1998). However, there remains another, more principal problem with this approach that has to be considered in greater detail: the variations inherent to the results of anytime algorithms may contradict the ultimate prerequisite of dead reckoning (using the *same* predictive algorithm).

AN OUTLOOK ON FUTURE RESEARCH

The most promising perspective of content-based reckoning opens if we consider again the main reason to set it up originally: the danger of dissociation, which so far is reduced but not banned. Encoding game states into semantic concepts that cover longer time intervals allows us at least to recognize earlier the danger of dissociation. So far, we have used the server only to forward the correction messages. It can also maintain its own local copy of the game world predicting the behaviour of all PCs (no encoding of a player or comparison necessary here). Then, with a decoding of the concepts presently active that is faster than real (game) time, the server is able to anticipate the probable development of the game for some time ahead. Some of those game states may evoke strategic rules activating critical interactions of two players with particularly high latencies (it is easy to keep track of the current average latency to each client). For example, in the soccer domain, two characters of opposing teams may run toward each other from some distance – one having the ball, the other performing the beginning of an attack event. The server recognizes that the game is likely to dissociate when the *attack* concept is realized much further (because this initiates critical types of interaction beyond the potential of the current latencies), and may therefore initiate resource-sensitive countermeasures – if available.

In the Germanic world of legends, Wotan appoints the Valkyries to influence with minimal "perceptibility" for the participants the fights between human heroes, and to decide the encounters as he demands – based on his general plan and the earlier "performances" of the heroes. The legendary schema, which we name "the Wotan Principle", suggests an idea of how to keep together a dissociating game without too much loss of "gameplay": when it recognizes a probable dissociation, the server (acting as Wotan) decides how the sensitive interaction is going to happen, though only on a very general, coarse scale (essentially: the results). To this purpose the server can, for example, script a sequence of the semantic concepts described above. The decision could depend on some general game principles or, in a rather futuristic version, on players' personal profiles derived automatically from earlier encounters. It is left to the clients (acting as "Valkyries") by merging the script with the players' reactions to set up the events accordingly, giving the local player some room for her/his own activities (i.e., remaining relatively imperceptible) nevertheless binding him or her into the general schema harmonized *a priori* with the other players in focus. Of course, the concrete sequence of events may be realised in quite a different way for each of the players involved in a dissociating scene. But the overall results are unique and allow all players ("surviving" – in a shooter game) to continue after the dissociation in a non-dissociated game with relatively similar memories of that encounter. The effects of this hidden authority on the "gameplay" feeling are yet unknown and form a major focus of interest in our future research on content-based reckoning.

* * *

Whilst most computer games are said to be provided with an internal AI, few results from Artificial Intelligence research on visual event recognition, natural language generation, and user modelling of the past twenty years have had a noticeable influence on the development of computer games so far. The transfer described in this paper is an attempt to improve this situation. In particular, insights in the cognitive aspects of defining motion verbs and spatial relations may hopefully play a more prominent role in computer games in the future.

REFERENCES

André, E.; G. Herzog; and T. Rist. 1988. "On the Simultaneous Interpretation of Real World Image Sequences and their Natural Language Description: The System SOCCER." In *Proceedings of the 8th European Conference on Artificial Intelligence* 1988 (Munich). Pittman, London, 449-454.

Blocher, A. and J. R. J. Schirra. 1995. "Optional deep case filling and focus control with mental images: ANTLIMA-KOREF." In *Proceedings of the International Joint Conference on Artificial Intelligence* 1995 (Montreal, August 20–25). 417–423.

Halper, N.; R. Helbing; and Th. Strothotte. 2001. "A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence." In *Computer Graphics Forum: Proceedings of Eurographics 2001*. Manchester, 2001. To appear.

Herzog, G. 1992. "Utilizing Interval-Based Event Representations for Incremental High-Level Scene Analysis." In *Proceedings of the 4th International Workshop on Semantics of Time, Space, and Movement and Spatio-Temporal Reasoning* 1992 (Château de Bonas, France), M. Aurnague (ed.), IRIT, Toulouse, 425–435,

Herzog, G. and P. Wazinski. 1994. "VIsual TRAnslator: Linking Perceptions and Natural Language Descriptions." *Artificial Intelligence Review* 8 (2/3), 175–187.

Mauve, M. 2000. "How to Keep a Dead Man from Shooting." In *Proceedings of the 7th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS)* 2000, Enschede, The Netherlands, 199–204.

Mellon, L. and D. West. 1995. "Architectural Optimizations to Advanced Distributed Simulation." In *Proceedings of the 1995 Winter Simulation Conference*. Ch. Alexopoulos (ed.). IEEE, Piscataway, N.J., 634–641.

Schirra, J. R. J. 1993. "A Contribution to Reference Semantics of Spatial Prepositions: The Visualization Problem and its Solution in VITRA." In *The Semantics of Prepositions - From Mental Processing to Natural Language Processing 1993*, C. Zelinsky-Wibbelt (Ed.). Mouton de Gruyter, Berlin, 471–515.

Schirra, J. R. J. 1995. "Understanding Radio Broadcasts On Soccer: The Concept 'Mental Image' and Its Use in Spatial Reasoning." In *Bilder im Geiste: Zur kognitiven und erkenntnistheoretischen Funktion piktorialer Repräsentationen*. K. Sachs-Hombach (Ed.). Rodopi, Amsterdam, 1995, 107–136.

Schirra, J. R. J.; G. Bosch; C.K. Sung; and G. Zimmermann. 1987. "From Image Sequences to Natural Language: A First Step towards Automatic Perception and Description of Motion." *Applied Artificial Intelligence* 1(3), 287–305.

Schirra, J. R. J. and E. Stopp. 1993. "ANTLIMA – A Listener Model with Mental Images." In *Proceedings of the International Joint Conference on Artificial Intelligence* 1993 (Chambéry, France, Aug. 29 – Sep. 3). 175–180.

Van Lent, M. and J. Laird. 1999. "Developing an Artificial Intelligence Engine." In *Proceedings of the Game Developers Conference* 1999 (San Jose, Ca., Mar. 16–18). 577–588.

Wahlster, W.; A. Blocher; J. Baus; E. Stopp; and H. Speiser. 1998. „Ressourcenadaptierende Objektlokalisation: Sprachliche Raumbeschreibung unter Zeitdruck." *Kognitionswissenschaft* 7(3), 111–117.

Zilberstein, S. 1996. „Using Anytime Algorithms in Intelligent Systems." *AI Magazine* Fall 1996, 73–83.