Dynamic Scene Description System — ODYS

Jacek Malec
Department of Computer and Information Science
Linköping University
S–581 83 Linköping, Sweden
jacma@ida.liu.se

May 13, 1996

Abstract

In the paper a system for analysing and describing dynamic scene of a robot is presented. The system, called ODYS (Opis DYnamicznej Sceny robota), is able to analyse descriptions of consecutive frames of the robot's environment, and to form on this basis high level descriptions of perceived objects' behaviours. The process of forming the description (and predicting the object's behaviour) is based on constructing frames called *events* and *scenarios*, which form a four level *hierarchy of description specificity*. Frames at each level describe respectively: all the changes that were recognized on the scene, changes considered according to some criterions as important, collections of simultaneously occurring changes, and series of changes linked with each other by the cause—effect relationship. Events and scenarios belonging to higher layer of description specificity form more general and more concise descriptions of the robot's scene than events and scenarios of the layer below. The system presented in the paper is coded in a dialect of COMMON LISP, and partially debugged and running.

1 Introduction

In the paper a system for analysing and describing dynamic scene of a robot is presented. The system, called ODYS (Opis DYnamicznej Sceny robota) is able to analyse descriptions of consecutive frames of the robot's environment, and to form on this basis high level descriptions and description predictions of perceived objects' behaviours. The process of forming the description consists of constructing appropriately chosen knowledge structures, and on analysing them in the way allowing extraction of information looked for.

The knowledge representation method described in the first part of the paper, is based on two notions: *event* and *scenario*. The event frame, being description of elementary change of the environment state, is built separately for every object recognized on the scene, and for every increment of the system's time. The scenario frame is created for every object, and is an aggregate of knowledge about object changes collected through longer period of time.

For the purpose of describing dynamical scene of an intelligent robot, a four layer *hierarchy of specificity of descriptions* is introduced. Events and scenarios, although of different kinds, are created on every level, and describe respectively: all the changes that can be recognized on the scene, changes considered according to some criteria as important, collections of simultaneously occurring changes, and series of changes linked with each other by the cause-effect relation. Events and scenarios at the higher layer are more general and more concise descriptions of the robot's scene than events and scenarios at the layers below.

Our method makes possible representing knowledge about arbitrary scene changes, either caused by the intelligent system itself, or independent of the will and actions of the system. It is an example of procedural representation extensively using the frame as the basic data structure.

The proposed knowledge representation method is used by the system ODYS, being the principal element of currently developed control system of an autonomous intelligent vehicle. The second part of the paper contains description of the system (ie. its input/output characterization and its position in the whole control system). The paper ends with the summary of results, description of the current status of the project, and proposals for the future research.

2 Knowledge Representation

For the purpose of describing dynamical scene of a robot an original knowledge representation method has been developped. The method is direct descendant of frame methodology of representing knowledge, and of procedural approaches to this problem. In the first part of this section the chosen knowledge structures are described. Second part is devoted to presentation of the four layer hierarchy of description specificity. The last part contains the discussion why this particular approach has been chosen, remarks about its (un)originality, and comparison with other solutions.

2.1 Knowledge Representation Structures

For the purpose of representing knowledge about dynamical environment of a robot, frame structures have been chosen. Depending on the type of input data they describe, they will be further called events or scenarios.

Event is a description of elementary change of behaviour of an object. As the time is assumed discrete, then for any given interval of time, and for an arbitrary object, only finite number of changes can be perceived, and then only finite number of events must be created. Under this assumption no information is lost during the process of creating event frames. Two different objects always have different events as descriptions of their changes, and so an event is always associated with particular object and particular point (or elementary interval) in time.

Scenario is a description of series of events somehow linked together. In most cases, this is the object being described, what links events forming a scenario, but this rule is valid only on the lower levels of scene description hierarchy. Scenario can be seen as description of an object's behaviour in longer period of time.

The lowest level structures are derived from information obtained directly from the scene analysis subsystem. If it is a vision system, a lot of data can be gathered and included in the event frame (called on this level the *elementary event*). Typical elementary event contains four slots: *object* being described, *time* (moment) of the event, *type of the event*, and its *description*. Descriptions are frames containing detailed information about the elementary change. The type slot is filled by symbolic description of an event, taken from predefined dictionary of descriptions. For the purpose of describing changes occurring on the scene of a robot, a simple type hierarchy has been proposed in [9].

The *description* frame is in the most simple case a list of differences between values of object's attributes in two adjacent time points (input frames). If it has sense, this difference is further processed to obtain qualitative or quantitative characterization of this difference, e.g. on the basis of velocity values the acceleration is computed and stored for the future needs.

As on this level only at most one event for one time point is created for an object, the scenario of this object is a straightforward aggregation (through the time-line) of the data stored in elementary events describing elementary changes of that object. The *movement/change scenario* frame has slots characterizing the scenario itself (*object, list of elementary events, starting time, ending time*), as well as the changes being described (*qualitative description, quantitative description*, and *predictions* of both). The predictions created here can be used as a feedback to the input (e.g. vision) system to allow faster processing of consecutive input frames.

Elementary events and movement/change scenarios, described briefly above, form the lowest level of description hierarchy. In the next subsection the contents of higher level structures will be presented. The reader interested in details can find them in [11] (its english version is in preparation); [9] contains more detailed description of knowledge structures used later, with several modifications, in ODYS.

2.2 Description specificity levels

The basic idea of proposed method lays in introducing simple and similar knowledge structures, but powerful enough to describe complex behaviour. Next three levels of description specificity use also *event* and *scenario* frames to represent information about the scene, but these structures contain on every level more general and more concise descriptions than on the level below. Events on the first level are used to describe elementary changes of physical object; on every higher level they are descriptions of elementary changes of *conceptual objects*: namely of the events and scenarios from the level below. The higher levels scenarios are also created around the conceptual objects (namely around their changes), although they may contain slot named "object" pointing to the physical object perceived on the scene. In the rest of this paper by *object* we mean physical object, whose behaviour is being described.

On the second level we intended to obtain descriptions of *substantial* (in some sense) changes of an object's behaviour. Being more precise, under term "substantial" we mean such changes that have influence on qualitative parameters of a movement/change scenario. We also have wanted to isolate changes that were either unexpected or not predictable.

Events on this level are called *change events*. They include descriptions of either unexpected changes of object's description detected in movement/change scenario, or changes that lead to different (qualitatively) behaviour of an object (e.g. object stops, changes direction of movement, changes shape, etc.).

Substantial change scenarios are created from change events, and are series of descriptions of substantial changes of an object. The slots of "level 2" frames are analogical to their counterparts in "level 1" structures. For example, a change event has the following slots: *subject* (i.e. movement/change scenario), *object, time, type*, and *description* (a frame structure).

"Level 3" structures are called *simultaneity events* and *cause-effect scenarios*. Frames on the first two levels were centered around an object and were its descriptions. On this level events are "time-oriented", i.e. substantial changes of different objects, but occurring in the same time point (or elementary time interval), are linked (described) in the knowledge structure. The reason of creating such structures is to find *simple* (in the sense: non delayed) *causal relations between objects' behaviours*. Of course, it is not true that every temporal coincidence of substantial changes means direct causal dependence, but rejection of such hypothesis is not justified without analysis.

Simultaneity event has four slots: *subject* — list of these substantial change scenarios, in which changes occur in the same time point, *time, qualitative description* — composition of substantial changes of parameters describing objects' behaviours, and *quantitative description* — list of differences between these parameters' values. Because these dependences can be different, one simultaneity event is created for every type of difference. E.g. if in the same time one object stops, second one starts moving, and the third suddenly changes direction of movement, three simultaneity events are created, one for each of the three possible pairs.

Cause-effect scenarios are again created for every object. They have, beside usual slots: *subject*, *object*, *list of simultaneity events*, *starting time*, *ending time*, *qualitative and quantitative descriptions*, two slots more: *hypothesis* and *cause*. A non-trivial amount of knowledge and reasoning is needed to fill these slots with respectively, the most plausible explanation(s) of observed temporal coincidence, and the most probable cause(s) of such coincidence. The value of these slots are formulated in first order predicate language, where possible predicate names are predefined.

The structures of fourth (the highest) level are called *temporal dependence events* and *process scenarios*. Events are created after discovering in one or more cause-effect scenarios one of the following dependences:

- *cyclic* occurrence of either a sequence of simultaneity events in a cause-effect scenario, or a sequence of change events in a substantial change scenario;
- either *OVERLAPping* or *MEETing* (in the sense of [1]) of time extents of two or more cause-effect scenarios;
- *functional* (computable) or *qualitative dependence* between quantitative descriptions of two cause-effect scenarios.

The dependence events have common structure: *list of scenarios*, *starting time*, *ending time*, *type*, and *description*. Number and contents of slots of the description frame depend of the type of created event.

The process scenarios have the following slots: *list of events*, *starting time*, *ending time*, *explanation*, and *description*. The description frame (in great part borrowed from [6]) consist of: *object list* (with distinguished, whenever possible, *agents*, *instruments*, and *subjects*), *process starting conditions, process continuing conditions, process ending conditions, interrupts* (possible behaviours — processes — after interrupting current process), *similar processes, temporal* and *causal clocks, list of process' models, unavoidable effects,* and *possible effects.* Due to the lack of space we do not describe the process scenario further. We hope that the slot names are intuitive and self-explanatory; interested readers can find detailed descriptions in [11].

2.3 Discussion

Choosing appropriate knowledge representation structures for a given problem (in our case gathering knowledge about dynamical scene of an intelligent robot) is the first, and the most important decision point in AI system design. We have chosen frame structures (called here events and scenarios) because of several reasons. First, there are many tools for creating and processing frame structures, beginning from COMMON LISP, and ending on specialized frame languages like e.g. KRL. Second, our idea was to use simple structures to allow the system to handle them easily, yet powerful enough to express complex relations at every level of description — starting from raw data, and finishing on causal dependences. Third, frames suggest intuitive partitioning of knowledge into small, easy to use primitives. Fourth, they allow simultaneous use of declarative and procedural types of knowledge. And fifth, they do not impose any restrictions on reasoning subsystem using them as knowledge structures — its control part can be designed with maximal clarity and effectiveness.

One of the reasons why we have chosen the process scenarios as the highest level of description is the following: it seems that the conscious part of the processes which we would like to simulate (recognizing, planning, reasonable acting), human beings perform only on causal level. E.g. if I (build a) plan to go to my office, the result seems like that: *go to the car, drive to the center, drive to the university, park somewhere, walk to the office*. No time, and only few spatial, constraints are involved — only the processes of walking and driving. Other constraints depend on the actual state of affairs during plan execution — if there are rush hours, if I have got enough fuel, etc. If one wants to simulate such way of reasoning and behaving, it seems that process scenarios are proper knowledge structures to capture the results of scene analysis as well as to serve as a basic element for plan formation.

As frames are extensively used for more than ten years, we do not claim our knowledge structures original. However, we would like to stress the fact of introducing the four level hierarchy of similar knowledge entities, with similar algorithms needed to create and analyse them, and completely separated computationally. It means, that information on every level can be processed independently of the previous results of higher levels of analysis. This is another source of possible processing speed-up (through pipelining), after parallelisms inherent for every of four levels of description specificity, where computations can be led independently (with some exceptions) for each object recognized on scene.

Another advantage of frame structures is their "user-friendliness". As quite a big amount of knowledge has to be built into the system before we could expect reasonable (valuable) output (especially levels three and four need this knowledge, and of course tools to process it), it should have been easy to code this information without effort counted in dozens of man-years. As it will be shown in the next chapter, the proposed knowledge structurization allows to create independent inference modules, able to work in parallel, and less complex than, e.g. a global rule-based inference system, to mention only the solution the most commonly used nowadays.

Our proposal of knowledge representation method is direct descendant of ideas of the early paper [6]. Many papers should be listed as inspiration for this work, however we can only mention these directly influencing our research. The problem of describing dynamic scene, and its early solutions, were stated in [6, 2, 20, 19].

Scripts were proposed in [18], and their use as knowledge representation tool for real-world scenes was analysed in [3, 14, 15, 17]. The description (or processing) hierarchy was introduced e.g. in [5, 4]. Comparing the proposed method with these and other solutions, we have not found any with the same properties, as ours method have. The closest one is that from [5], but there the manually created process descriptions have to be used, where in our case they can be generated by the system itself.

3 The System

This section contains the description of the ODYS system. First, we show the place, that ODYS is to occupy in a control system of an intelligent robot. Then, we give its input/output specification, and briefly describe reasoning performed on each level of scene description specificity. At the end we describe processing of a very simple example.

3.1 The control system of an autonomous vehicle

The architecture of a control system of an intelligent robot is presented on fig. 1 ([8, 10]). The final version of ODYS consists of lower levels of the REASONING subsystem, i.e. the modules realizing *short-* and *long-term predictions*, *scenario analysis*, and *reflexes*, and on the next layer the *learning*, *scenario creation*, and *memorization modules*. But the memory module itself is external to our system — ODYS does not perform any memory management functions like indexing in long-term memory (LTM), or retrieving information from LTM to STM.

On its input ODYS obtains descriptions of consecutive frames (images) of robot's scene. A description is a frame structure containing the following slots: time (i.e. the frame number), list of objects recognized on scene, and for every object from that list a slot containing the object's name (given by ODYS), and a pointer to the frame structure containing description of that object in given moment of time.

The object descriptions have the form resembling that from [3], with few minor differences. They are frame structures being either instances of the object's model frame (if the model is known to the system), or the most accurate description of the object (if the system cannot calssify it to one of the classes it has known).

We assume that the vision module is able to generate such descriptions in the real-time. This assumption is of course very strong, however the progress in the computer vision field makes it justified. Currently the description frames are created artificially.

ODYS outputs the process scenarios, which are ready to be used by a high-level reasoning (e.g. planning) module. Its work can be seen as cyclic creating of

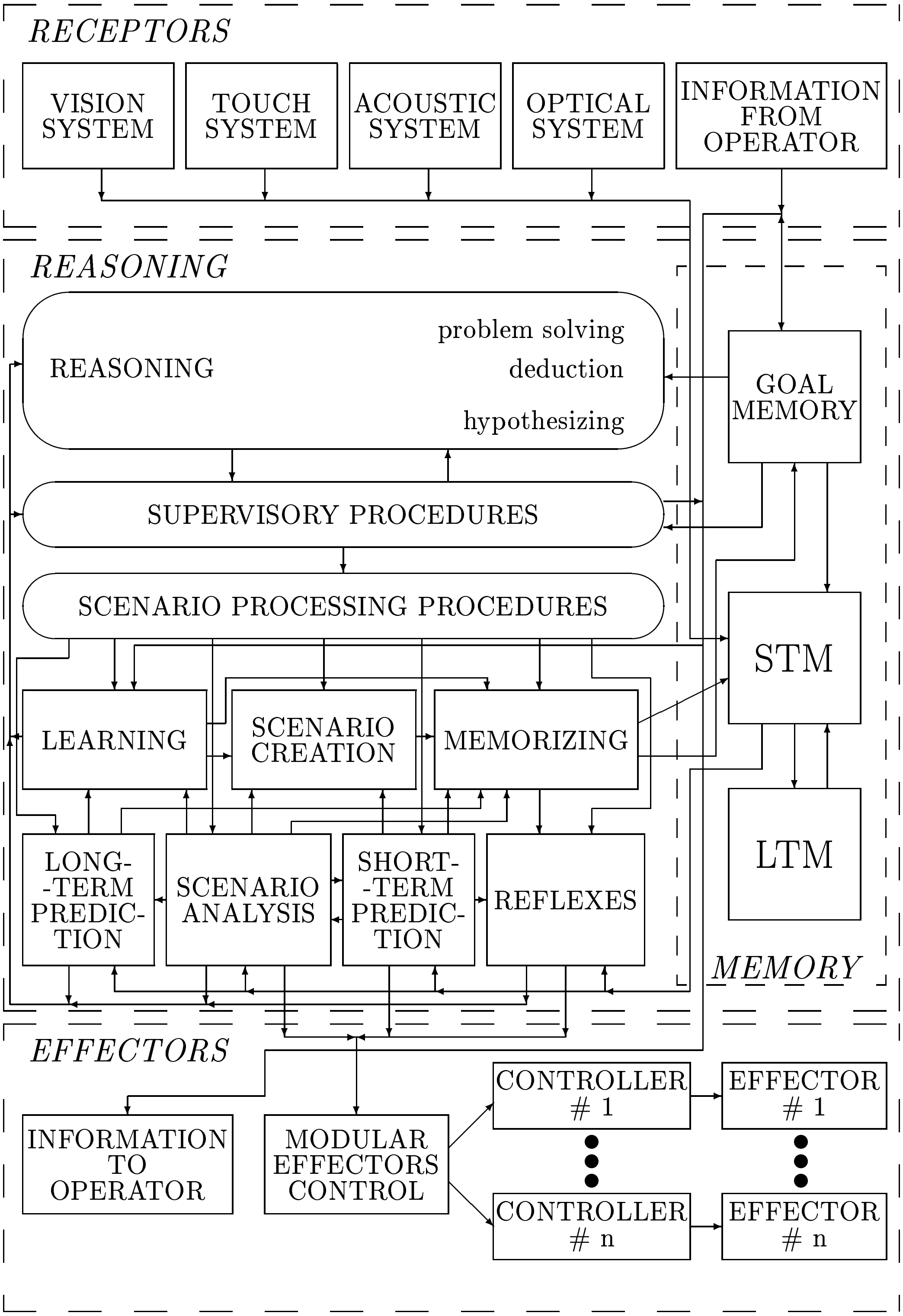

Figure 1: Architecture of the control system of an intelligent robot

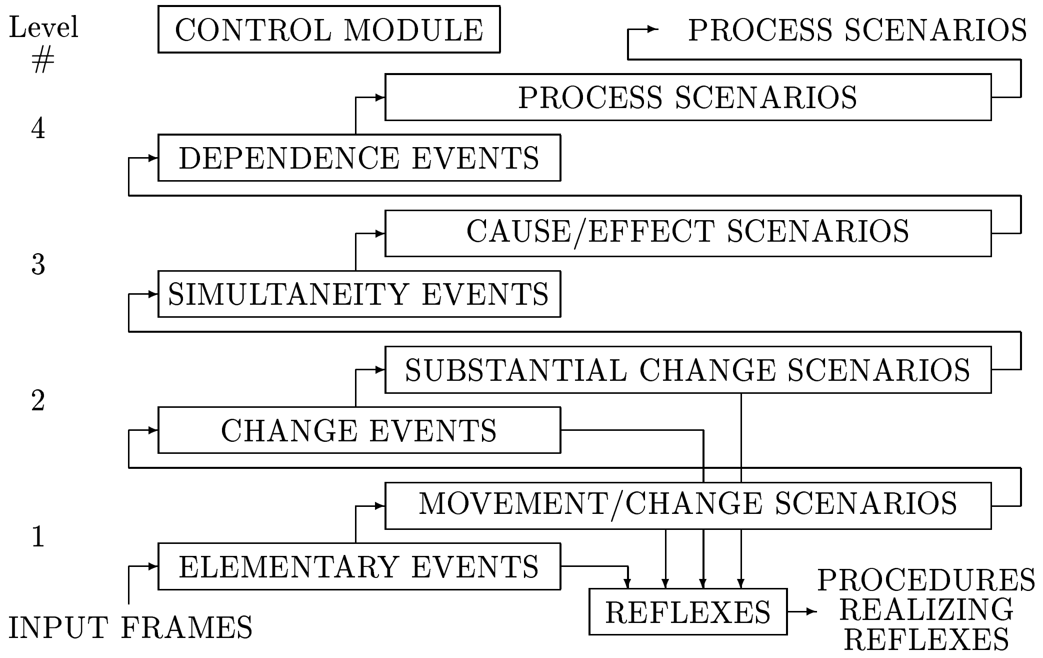

Figure 2: Structure of the realized part of ODYS

events and updating the scenarios describing behaviours of the objects recognized on scene. As the intermediate results of the ODYS' work (lower levels events and scenarios) are stored in memory, they are also available during later stages of reasoning, outside ODYS.

3.2 Description of the system

Current version of ODYS consists of fully designed algorithms for the lowest level part of the control system of an autonomous vehicle (fig. 1) — scenario analysis demons, short- and long-term predictions, and reflexes. Learning and memorization modules are in the developpement stage.

The designed modules of the system form the structure shown on fig. 2. Among them, algorithms of levels 1, 2, and 3, and the control module are coded and debugged. The algorithms of scenario analysis and prediction are coded as demons invoked by changes of the appropriate knowledge structure of the lower level. However, as ODYS is implemented in Common Lisp on a PC, no parallelism could be introduced, and consequently, the control module is rather simple in the coded version.

3.3 Example

As an example we have chosen a simple scene, on which an object appears, deccelerates (under the friction force), and finally stops. After some time appears an arm of a manipulator, approaches to the object, grips it, and takes away outside the visibility region of the camera.

Thirty image descriptions have been generated and feeded to ODYS. The system has generated 109 elementary events, 115 change events, 65 simultaneity events, 7 dependence events, formed 6 movement/change scenarios (four for each part of the manipulator arm, fifth for the arm as a whole, and sixth for the object), 6 substantial change scenarios, 6 cause — effect scenarios, and finally 2 process scenarios — first for appearance of the manipulator arm, its approach to the object, and stopping, and second for gripping the object and taking it away. As the appearance of the object, and its behaviour immediately afterwards, is not causally connected with any other object, no process scenario is created for it.

4 Conclusions

Analysing and describing dynamic scene of an autonomous intelligent vehicle is the problem tackled in many recent research projects. However, the suggested (or realized) solutions do not introduce any methodology — they are direct implications of current state of the art in sensor technology , and applications of sensor fusion techniques.

Our central goal was to design and build effective system, easy to integrate with control system of an autonomous robot. So we also have used well-known knowledge representation blocks, and combined them into a system having expected properties. As neither the system, nor modules surrounding it, are completed yet, we cannot draw straightforward conclusions, like: the proposed method allows to effectively control an autonomous vehicle in real-time. However, the results of simulations done so far allow to suspect, that this direction of research might give valuable experimental results in the nearest future. In particular, we would like to obtain some suggestions concerning methodology of constructing such systems, and some insight into knowledge acquisition process.

The main purpose of this paper was to present an approach to knowledge representation problem, which can be exploited in real-time systems. The method itself is complete, and in our opinion, worth presenting in such extent. Of course, many questions are worth further investigations, or at least commenting. First, the mostly procedural and non-deductive approach to the problem — its justification are its effectiveness shown so far, and on the other side, lack of practical results of the deductive approach (commented e.g. in [13]). Second, the architecture of the system, being result of simple and rather technical considerations, resembles much the blackboard architecture [16]; it could be interesting to further investigate this similarity. Third, we are interested in reasoning processes, which underlay constructing descriptions of dynamic scenes (see [12, 14]), so stating and analysing explicitly such processes, which are embodied in ODYS, might be worth further research.

Introductory analysis of the algorithms constituting ODYS can lead to some conclusions about the knowledge elicitation process which takes place during formation of the output description. They will be listed below, including the obvious facts for completeness.

1. The knowledge to be elicited must be present in the input information stream — nonsensical data can give no (or nonsensical) results.

2. It must be known to the system, how to extract the desired piece of knowledge. In order to be able to learn effectively, the system must possess a big amount of procedural knowledge allowing extraction of information looked for (see [7] for much deeper analysis of this problem).

3. If the goal is to obtain detailed description of the process perceived on scene, the system must have appropriate conceptual structures able to express details of perceived events. Describing (the scene) can be seen as classifying — the more complex (deep and wide) the prototype net (or tree), the more detailed the resulting description. This remark is also valid for systems able to learn about new prototypes — such systems have to have "place" for this prototype descriptions, i.e. the prototype net must also be deep and/or wide with, possibly, empty nodes to be filled during the learning process.

4. Multi-level organization of knowledge processing modules allows to simplify the resulting procedures — instead of one, huge program one obtains a chain of smaller and simpler ones (an example of such organization can be found in [5]).

5. The process of knowledge elicitation should be made granular, that is, should be decomposed into many independent procedures, whose effects can be composed back into piece of knowledge qualitatively different from the input information.

6. The more domain-specific the knowledge processing algorithms, the faster the processing itself, but the narrower the application area of the algorithm ([14, 12]).

Our plans connected with the implementation side of the system include completing design of the higher layer algorithms, coding them, and simulating parallelism of demon invocation in order to estimate the ODYS' ability to work in the real-time. Final stages of assembly of an autonomous vehicle platform, and on the other side, of a hardware vision processor, allow to plan the full-scale experiments in predictable future. We would also like to further investigate the problems of knowledge acquisition and knowledge elicitation which appear during dynamic scene description.

5 Acknowledgements

The research described in the paper was conducted partially under the grant CPBP 02.13 from Polish Academy of Sciences, Institute of Fundamental Problems of Engineering. Its results were presented as the author's PhD Thesis [11]. The author is indebted to the thesis supervisor — prof. J. Jaroń, for his encouragement and help, and to the referees — professors L. Bolc, C. Daniłowicz, and R. Jakubowski, for their valuable remarks. Also the continuous help of my colleagues from the Cybernetics and Robotics Group should be acknowledged; without it this work still wouldn't have been finished.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26, 1983.

[2] N. Badler. *Temporal Scene Analysis: Conceptual Descriptions of Object Movements*. PhD thesis, University of Pennsylvania, 1975. Report UP-MS-CIS-76-4.

[3] R. A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17(1–3):285–348, 1981.

[4] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2:14–23, 1986.

[5] P. A. Fishwick. *Hierarchical Reasoning: Simulating Complex Processes over Multiple Levels of Abstraction*. PhD thesis, University of Pennsylvania, 1986. Report MS-CIS-86-21.

[6] G. G. Hendrix. Modelling simultaneous actions and continuous processes. *Artificial Intelligence*, 4:145–180, 1973.

[7] D. Lenat, M. Prakash, and M. Shepherd. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottleneck. *AI Magazine*, 6(4):65–85, 1985.

[8] Jacek Malec. The control system of an intelligent robot acting on dynamic scenes. In *Proceedings ECAI-84*, pages 557–558, Pisa, 1984. Elsevier North-Holland.

[9] Jacek Malec. Scenarios as a tool for dynamic scene representation. In R. Trappl, editor, *Cybernetics and Systems Research 2*, pages 781–786. Elsevier North-Holland, Amsterdam, 1984.

[10] Jacek Malec. Architektura systemu sterowania robotem inteligentnym (An architecture of a control system of an intelligent robot). In *National Conference on Robotics, Wrocław*, pages 159–166, Wrocław, Poland, 1985. Wydawnictwo Politechniki Wrocławskiej.

[11] Jacek Malec. *Pewna metoda reprezentacji wiedzy o dynamicznej scenie robota (A Method for Representing Knowledge about Dynamical Environment of a Robot).* PhD thesis, Technical University of Wrocław, Wrocław, Poland, 1987.

[12] Jacek Malec. Reasoning underlying construction of the model of dynamical world. In L. O. Hertzberger and F. C. A. Groen, editors, *Intelligent Autonomous Systems*, pages 382–385, Amsterdam, 1987. North-Holland.

[13] D.V. McDermott. Critique of pure reason. *Computational Intelligence*, 1987.

[14] B. Neumann. Knowledge sources for understanding and describing image sequences. In W. Wahlster, editor, *Proc. GWAI-82*, pages 1–21. Springer, 1982.

[15] B. Neumann and H.-J. Novak. Event models for recognition and natural laqnguage description of events in real-world image sequences. In *Proceeding of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe*. Morgan Kaufman, 1983.

[16] H. P. Nii. Blackboard systems: An overview. *AI Magazine*, 6, 1985.

[17] G. Retz-Schmidt. The script-based generation and evaluation of expectations in traffic scenes. In H. Stoyan, editor, *Proc. GWAI-85*, pages 197–203. Springer, 1986.

[18] R. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[19] J. K. Tsotsos. *A Framework for Visual Motion Understanding.* PhD thesis, University of Toronto, 1980. TR CSRG-114.

[20] S. Tsuji et al. Understanding a simple cartoon film by a computer vision system. In *Proceeding of the Fifth International Joint Conference on Artificial Intelligence*, pages 609–610. Morgan Kaufman, 1977.