Control and Cybernetics

vol. **32** (2003) No. 4

Control aspects of the blackboard agent architecture for a mobile robot [1]

by

Grażyna Brzykcy, Jacek Martinek, Adam Meissner, Piotr Skrzypczyński

Institute of Control and Information Engineering
Technical University of Poznań
Plac M. Skłodowskiej-Curie 5
60-965 Poznań, Poland

Abstract: In the paper, the architecture of a mobile robot cooperating with other robots and some stationary devices in a task of collective perception and world modeling is considered. We present data-driven processing of information performed by an individual robot treated as an agent and we propose to organize it as a set of experts (also treated as agents) exchanging data by means of a blackboard. The roles of particular agents and the structure of the blackboard are described. We analyze the control aspects of the system and the form of control knowledge.

Keywords: mobile robot, agent system, blackboard system, control knowledge.

1. Introduction

Robotic systems, including mobile robots, can perform different complex tasks in such areas as production or transportation. However, regardless of the particular tasks solved by robotic system, the mobile robot behavior always strongly depends on the available model of the surroundings (the environment, the world) in which the robot is active. Creation of such a model is often performed in a dynamic way on the basis of data from robot sensors.

Autonomous mobile robots update their internal world models on-line in order to reflect the scene changes and the new evidence from sensors. In a multisensor system hosted on a mobile platform an integration of different physical sensors with various principles of operation and distinct abilities to extract features has to be performed. An efficient use of all information available in the robotic system and coherent fusion of the data from different knowledge sources are expected. Thus, collective perception and collective world modeling are needed. If every component of the system represents a different perspective and uses a different representation of data, and all components work cooperatively, then the solution can be found faster and the system is more reliable and more resistant to unexpected disturbances in the operation of its components.

This paper presents the results of research concerning the software architecture of a distributed, multi-sensor and multi-robot system that is aimed at the task of collective perception and world modeling.

2. The multi-robot system

In the proposed system, mobile robots with sets of sensors and some stationary devices, like a scene monitoring overhead (ceiling-mounted) camera, are separate components, Brzykcy et al. (2001b). Moreover a knowledge base, containing *a priori* given data about the environment, is available and may be supplied by the system operator, Kasiński et al. (1998). The system structure is not fixed. The number and the characteristics of components may vary, so the system is open to modifications. The particular components have different perception abilities and their individual world knowledge is limited, potentially incomplete, uncertain or out-of-date.

The collective perception and world-modeling task is performed by every mobile robot during its movement along an indoor route, which is described by a sequence of points. Information, gathered during this movement from own sensors of the robot, from other robots (on demand) and from stationary devices, is used to construct or to update the internal world model of the robot. The model meets the requirements of the robot navigation system.

In the prototype set-up, two mobile robots of *Labmate* type, and monitoring cameras are used. Both robots have on-board PC computers and are equipped with laser scanners and ultrasonic range finders (sonars). One of these robots is also equipped with a simple vision subsystem with one CCD camera. This on-board camera is solely used to detect passive artificial landmarks purposefully attached to objects in the environment, Kasiński et al. (2001). This assumption simplifies image processing and gives the vision subsystem a chance to contribute to the robot navigation task in real-time, by providing an alternative way of self-localization. The robot controller can gain data about the robot position and its orientation from odometry.

The preliminary analysis of the perception and world-modeling issues in the mobile robotics domain indicates the following properties of the system under study:

- components of the system act in an independent autonomous manner,
- all the components have a common area of operation (e.g. an industrial

- the robots and the environment influence each other: the actions of a robot (e.g. a displacement) depend on the state of the environment, and on the other hand the robot effects changes in its environment by executing actions,
- system components can communicate with each other.

To model a cooperative system with the above-mentioned properties the agent concept can be used, Hunhns et al. (1998). In the domain of robotics, agents are usually defined as autonomous or semi-autonomous hardware or software systems, which perform their tasks in a complex, dynamic environment, Müller (1996). Autonomy is understood here as the ability to make decisions based on an internal agent world representation, without being controlled by any central station. An agent has a perception and communication ability, and its functionality is expressed through the actions it takes, including the communication actions.

In the system under study the agent concept is used to model the mobile robot, the monitoring subsystem with ceiling-mounted camera and the human operator. The problem of the collective perception and world modeling in such a multi-agent system can be reduced to the organization problem of an effective co-operation among agents. The robot-agents play a particular role in this system, because they construct the world models. These models are stored in a form of vector- and raster-based maps, Kasiński et al. (1998).

In the reminder of this article we focus on a single robot-agent architecture and its control aspects.

3. Mobile robot architecture

3.1. Mobile robot tasks

During the process of perception and world modeling, the robot moves between given points in a partially unknown indoor environment. The robot observes the environment by means of available sensors and builds a model of it. We assume that when some obstacle is encountered on the preplanned path, the robot is able to detect it and to make a necessary detour. It is performed by means of a simple reactive navigation, Arkin (1998).

To perform the reflexive navigation an up-to-date representation of the surrounding environment is needed. This representation should be built fast and it should exploit all sensory information available to the robot.

An interesting proposal of local representation aimed at supporting obstacle avoidance is the occupancy grid built by using the Histogramic In-Motion Mapping (HIMM) algorithm, introduced by Borenstein and Koren (1991). Moreover, the grid-based map can be used as a common ground for fusion of different range sensor data.

The laser scanner can provide precise range measurements to the surrounding the scanner has a very small light spot, and due to this limitation it can overlook small obstacles (e.g. pipes, wires). The scanner can produce 2D map of the environment, either vector- or grid-based, Kasiński et al. (1998).

In contrary to laser sensors, ultrasonic range finders suffer from wide-beam problems and spectral reflections. But they can perceive obstacles not visible to the lasers (e.g. glass door), and having 3D (conical) field of view, can protect the robot from being "decapitated" by obstacles such as tables. The sonars are able to yield data for the grid-based map.

To build an environment model the robot needs information about its own position. This self-localization task is performed on the basis of data from odometry or by the matching of local and global environment models. A local environment representation is needed which can be matched effectively with the global environment model given *a priori*. The local vector-based map, which is composed of line segments corresponding to main geometric structures in the environment (e.g. walls), is such a representation. The localization task can also be solved by means of the on-board vision system or in cooperation with an overhead camera.

After the self-localization, the up-to-date world model is built by integration of data supplied by mobile robot sensors and data extracted from the *a priori* model of the environment. This model is provided to the mobile robot in the moment of system initialization. To compensate for its limited perceptual capabilities in the process of map building and to resolve the possible ambiguities in the model, the robot can exchange data with other agents in the system (e.g. other robots).

The global world model takes a form of the vector-based map. Unlike the local vector map it is structured in particular objects (sets of line segments) which resemble objects in the real environment. These objects are represented as polygons and poly-lines. They are attributed with some additional properties which are important to the model updating process, e.g. possibility to move an object or to modify its shape.

The robot should also have information about its route described by a sequence of intermediate points, and a list of artificial landmarks for the on-board vision system.

3.2. Data processing

In the mobile robot, a complicated, multi-stage data processing is undertaken:

- data can be provided by robot subsystems or they can originate from other agents of the system,
- different robot tasks are performed on the strength of data that are expressed in the diverse formats though they can have a common origin (e.g. the range information is used both in the raster and in the vector map),

- a vast majority of data processed in the system are the local data, i.e. they describe only a part of the environment,
- data have to be acquired and processed continuously while the robot is moving,
- the process of the world model building involves integration of very different types of information (sensor-based and *a priori*, local and global, uncertain and certain etc.).

Two kinds of actions can be distinguished in the information processing in the multi-agent system, namely the transformation of raw sensory data to the form of environment maps, and the exchange of data between different world model representations. Most of these processes have been recognized and described as the operators of the so called Perception Network for a group of mobile robots, proposed in Kasiński et al. (1998).

Operations described below are important with respect to the vital tasks of the mobile robot:

1. Estimation of the position and orientation (with uncertainty measure) from the robot odometry.
2. Local grid map update from the laser scanner data.
3. Local grid map update from the sonar data.
4. Local vector map building from the laser scanner data.
5. Analysis of the local grid map and the generation of the next move for the mobile robot controller (here a slightly modified Virtual Field Histogram algorithm from Borenstein et al. (1991) is used).
6. Conversion of the local grid-based map to the vector form.
7. Integration of the vector map extracted from the grid map with the local vector map generated directly from the scanner measurements.
8. Estimation of the current robot position and orientation by means of map matching.
9. Extraction of artificial visual landmarks from the environment by the onboard camera and the estimation of robot position from these data.
10. Optimal integration of all position estimates available to the robot at the given moment (including estimation from monitoring subsystem) by means of Kalman filtering, Skrzypczyński et al. (1999).
11. Global vector map update by using the current local vector map, current position estimate, and (possibly) pieces of vector maps from other mobile robots.

All these operations are well-determined independent subtasks that robot has to perform to achieve its goal (solve the problem). Each of the subtasks can be separately defined as a "black box" with some input and output. The "black boxes" are loosely coupled by data they exchange - the input of one "box" is the output of another. This kind of subtasks interaction can be realized via shared

The order of subtask execution is, however, not known in advance - the decision to perform a particular subtask is made dynamically using current data values (e.g. inconsistency of the position and/or orientation estimate triggers on-board landmark recognition system). This kind of data processing can be organized as a blackboard system, Engelmore et al. (1988) with a shared database and a set of "experts" cooperating in a data-driven and opportunistic way. A blackboard system can be regarded and implemented as an agent system, Schwartz (1995) with "experts" working as agents.

3.3. Multi-agent blackboard architecture of the mobile robot

3.3.1. Blackboard system

Some proposals of the blackboard architecture applications to multi-sensor system for an autonomous single robot were discussed in literature (e.g. Kappey et al., 1994). With relation to the task of collective perception and world modeling an idea to impose blackboard architecture on the robot-agent was sketched for the first time in Brzykcy (2000) and developed in Brzykcy et al. (2001b).

The blackboard system consists of three basic components:

- the data structure (blackboard) that is appropriate for the problem solving domain and is mostly organized as one or more application-specific hierarchies,
- the set of processing modules (knowledge sources, experts) that transform the data from the blackboard,
- the control machinery realized by a special expert.

The modules are kept separate and independent and each of them is able to perform an action. They join in solving the problem according to the following cycle:

- triggering the module in view of new information on the blackboard,
- recognition of information context (satisfaction of action preconditions),
- execution of an action,
- storing data in the blackboard.

The control mechanism is responsible for the execution of each problem solving cycle, particularly for allocation of processing resources to the most promising expert. Explicit representation of control facilitates the definition of the complex control strategies by the system users.

The blackboard system can be easily modeled as an agent system, Schwartz (1995). In the system under study both the processing modules and the information providers - sensors - placed around the blackboard are good candidates to be agents. Their architecture is potentially very simple because:

- each module has precisely defined tasks (e.g. feature transformation, data

- the blackboard (often the part of the blackboard) is an environment for each module; it is the source and destination of the module data,
- a module needs no knowledge about other modules.

This multi-agent blackboard architecture (MAB) of the system introduces additional advantages such as parallel data access (here blackboard data access) and concurrent execution of many tasks (agents). It is easy to introduce changes to the blackboard system as putting an agent in and deleting it from the system is very simple (this property is material in case of sensor set modification). Finally, the system organization does not depend on agent implementation.

3.3.2. Blackboard structure and collection of agents

In the blackboard system there is a strong dependency between the blackboard data structure and the repertoire of agents gathered around it. For the system designer the most desirable and effective way of acting is to obtain the blackboard structure that is a good model of the problem under study. In order to define the appropriate blackboard structure one has to establish what constitutes the desired solution. The designer ought to define also the necessary data and the knowledge how to process them. The abstraction levels of data representation on the blackboard during the process of problem solving determine the granularity and the way of knowledge division into agents that perform the separated subtasks. The higher number of intermediate levels, the more specialized processing agents. However, the way the problem is partitioned into sub-problems makes a great deal of difference to the clarity of the design, the efficiency of problem solving and the ability to solve the problem at all.

The set of agents - executors of subtasks proposed in Section 3.2 - is derived from the list of Perception Network operations, Kasiński et al. (1998) and seems to match well the problem of translating multi-sensor data, domain knowledge and *a priori* knowledge into aggregated world model. The blackboard structure and actions of particular agents are depicted below. The list of potential actions performed by a mobile robot in the course of collective perception and world modeling is not exhausted. Especially the subtasks originated in the problem of robot movement planning are not described. But the blackboard architecture does not restrict the possibility of enlargement of an action set; both the blackboard structure and collection of agents are easily modified.

The blackboard usually contains universally accessible data and in our system it contains different descriptions of the robot environment. The most primitive environment model takes the form of a local map. There are grid-based and vector-based local maps on the blackboard (*GridMap* and *LocalVectorMap* respectively). The grid map represents environment as an array of cells, each one holding a confidence that it is empty or occupied. This map is built by employing the HIMM algorithm, Borenstein et al. (1991). The vector-based map consists of line segment primitives described by vectors of parameters. Informamap (*VectorMapfromGrid*) is obtained from a grid-based one, then both of the vector maps are combined and at last the most abstract and extended model (*GlobalVectorMap*) is formed or updated. Information about the robot current position and orientation is also contained. The odometry (*OdoPos*), the on-board vision subsystem (*CamPos*), and the vector map matching procedure (*SensPos*) derive these data. They are unified into an optimal estimate (*CurrPos*) by the position comparison/fusion algorithm.

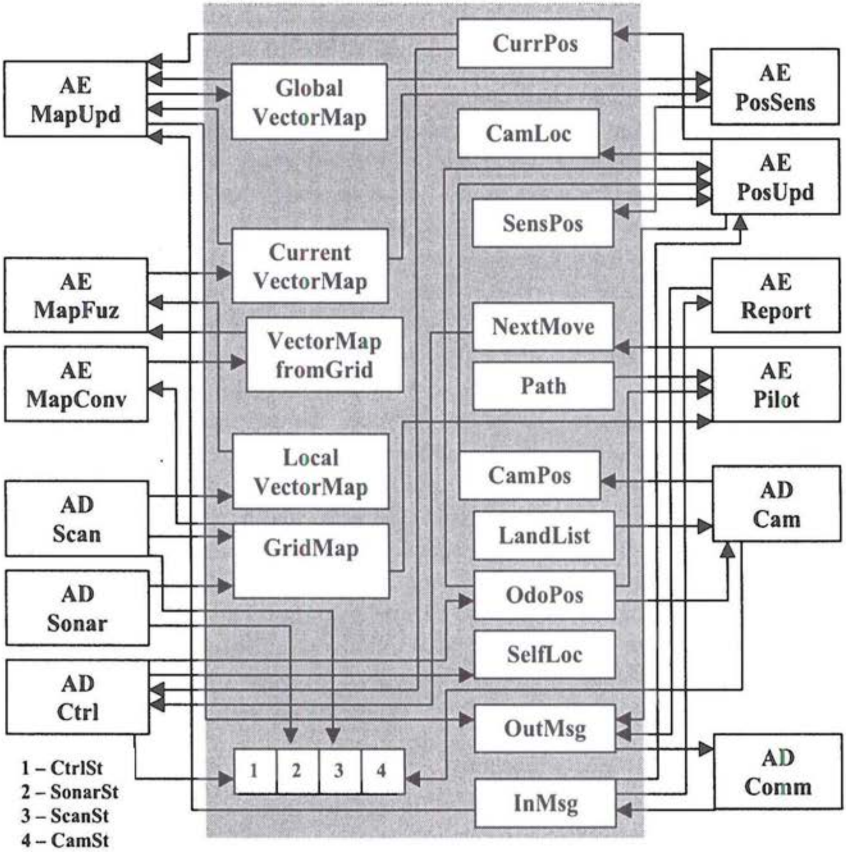

Figure 1. The MAB architecture of a mobile robot

The blackboard holds also elementary status data of individual sensors (e.g. *ScanSt*), necessary to indicate the states of devices (e.g. failure), the queue of outgoing messages (*OutMsg*), that are addressed to other agents (ceilingcoming messages (*InMsg*). Auxiliary data are connected, for instance, with the robot movement task (*Path*) or with the vision system operation (*LandList*) and information needed to arrange control for blackboard system, such as the self-localization task execution necessity (*SelfLoc*). Fig. 1 depicts the blackboard structure and the collection of agents. The main data flows are represented by means of arrows. To keep this illustration readable some of the links are neglected.

Agents in our system are related to physical devices - sensors and actuators (*ADxxx*) or to processing tasks - experts (*AExxx*). The device agents execute their actions concurrently, while preserving time constraints of respective sensors and actuators.

Around the blackboard there are sensor agents for sonars (*ADSonar*, action 3), the laser scanner (*ADScan*, actions 2 and 4) and the on-board camera (*ADCam*, action 10), together with the robot controller agent (*ADCtrl*, action 1) and the communication agent (*ADComm*). The scanner agent (*ADScan*) preliminarily processes data obtained from the scanner. This procedure is executed every fixed period of time or on demand. It processes the input data, updates a local grid map (*GridMap*) and builds a new local vector map (*LocalVectorMap*).

In order to update a local grid map the sonar agent (*ADSonar*) processes the data coming every pre-determined period of time from ultrasonic range finders.

The vision subsystem agent (*ADCam*) is activated when the current position and orientation of the mobile robot is needed. To achieve this goal the agent processes the acquired image, detecting landmarks and comparing them to the a priori given list of landmarks. The agent also may obtain the current robot position (*CurrPos*) from the blackboard.

The data gained from odometry, i.e. position and orientation of the robot (*OdoPos*), are cyclically stored on the blackboard by the agent of the robot controller (*ADCtrl*). Detection of obstacles on the robot path and the planning of future robot moves are tasks of the pilot agent (*AEPilot*, action 5).

The transition from the grid-based world model (*GridMap*) to vector-based model is performed by the map conversion agent (*AEMapConv*, action 6). In this case, the geometrical interpretation takes place at the much later stage than for the vector map obtained in direct line-based interpretation, and thus much more evidence can be accumulated, Skrzypczyński et al. (1999).

The local map fusion agent (*AEMapFuz*, action 7) compares a local vector map (*LocalVectorMap*) and a vector-based map acquired from the grid-based map (*VectorMapfromGrid*) in order to integrate the data of the same representation. The unification of maps comprises individual segment matching. This procedure is performed to build a new current vector map (*CurrentVectorMap*) every time a grid based map is transformed to a vector-based form.

In order to determine the position and orientation of the robot (*SensPos*), the self-localization agent (*AEPosSens*, action 8) compares the current vector map (*CurrentVectorMap*) and the global vector map just after its creation by the position update agent (*AEPosUpd*, action 9) with regard to data coming from such sources as odometry (*OdoPos*), the vision subsystem (*CamPos*), the vector map matching procedure (*SensPos*) or from other agents (e.g. ceiling-mounted camera).

The creation of a new vector map (*CurrentVectorMap*) and determination of the current robot position (*CurrPos*) are necessary conditions for initiating the process of updating the global world model (*GlobalVectorMap*). Maps are integrated by the global map update agent (*AEMapUpd*, action 11) which tries to unify both maps using additionally the domain knowledge. The agent evaluates the segment parameters and the attributes of the segment sets. It also detects situations when a robot has to acquire a part of the world model from another agent. Monitoring of the whole data and the operator command execution is due to the report agent (*AEReport*).

4. Control aspects of the architecture

In a blackboard system many agents perform computations and the specialized knowledge of each of them ought to be used when it is most desirable. On the one hand the moment, when an agent enters into the computation process, depends on the agent and on the other hand - on the system's willingness to take advantage of this agent knowledge. In a blackboard system, information about the readiness of both elements (agent and system) constitutes only a part of control knowledge. It is combined with the question (decision knowledge) which agent to choose from the set of operational agents.

Knowledge about willingness of particular agents has to originate from them because they are the domain experts and they know best when to start their actions. This moment is strongly relevant to the problem solving state. As the solution emerges on the blackboard, an agent observes this universally accessible data structure. Usually only some data or rather the change of their state (e.g. insertion, modification, or deletion) constitute the subject of the agent's interest. When, for instance, the self-localization flag (*SelfLoc*) is set, many agents will declare their willingness to act. The scanner agent wants to prepare a new local vector map of environment, the conversion agent - to transform a grid map into a vector one and the sonar agent - to suspend its continuous actions until the robot's current position will be known.

The decision to initiate the action of some agent depends on the problem solving strategy and on physical capabilities of the system. If it is possible to perform many actions at once, then perhaps each of the operational agents will initiate its activity (provided that the consistency of universally accessible data is provided). However, the operational agents are usually evaluated in view of their usefulness in the current problem solving state and only a subset of them is chosen (e.g. just one of them). This control problem (Hayes-Roth, 1985) is solved by a specialized agent, which is equipped with adequate control some new agents declare their willingness to act.

All the changes of particular data states that are vital for an agent behavior (agent initialization events) have to be specified in the system but the description may be stored at different places. Conventionally, it is assumed that the specification of events comprises a part of an agent definition. Consequently, the whole information about an agent is located at the same place, but the domain and control knowledge is amalgamated and control knowledge is distributed. Other proposals are known (e.g. Schwartz, 1995), where domain and control knowledge are separated and all agent-event dependencies are stored in the distinguished blackboard area. The control agents with their data are also kept there. This form of knowledge organization has some advantages: all the agents are uniformly treated by the system and this enclosing of control knowledge allows to define the complex problem solving strategies.

The control component of the mobile robot blackboard system cannot be complicated - the reduction of the processing time is a real challenge. Accordingly, the device agents and the pilot agent execute their actions concurrently, while preserving time constraints of respective sensors. It is the necessity of determining the robot's current position that triggers the remaining agents. Their actions have to be undertaken sequentially one after the other (except for two actions, which create local vector maps from the grid map and from the scanner data, and which can be performed concurrently); thus in our system the choice of an agent is a quite trivial decision. We may assume that each agent, which is ready to act, initiates its action. Now it remains to detect the events in the system and to inform appropriate agents when these events occur. In software environments, there are different mechanisms serving to notify an agent about events (e.g. the Delegation Event Model in Java, Bigus, 1998) that one can use to achieve this goal. All that an agent has to do is to declare what changes of what data it is interested in - the environment is responsible for notifying all the concerned agents.

The main events of our MAB architecture of the mobile robot system, the triggered agents and the actions, that are performed, are presented in Table 1.

5. Implementation of the architecture

All the most important modules of the MAB architecture have been implemented and integrated in Linux. Because the task of the prototype system is the co-operative, multi-sensor world modeling, it has been decided to centralize the strategic-level planning, and to integrate it with the user interface agent, Kasiński et al. (2002). The global route planning method is based on simplified Voronoi diagrams and the A* algorithm for optimal path determination. The global vector-based map is used as the world model for planning, Brzykcy et al. (2001b). The deliberative elements of the architecture, concerning global path (route) planning and robot task scheduling are not integrated with the robot the MAB software are presented.

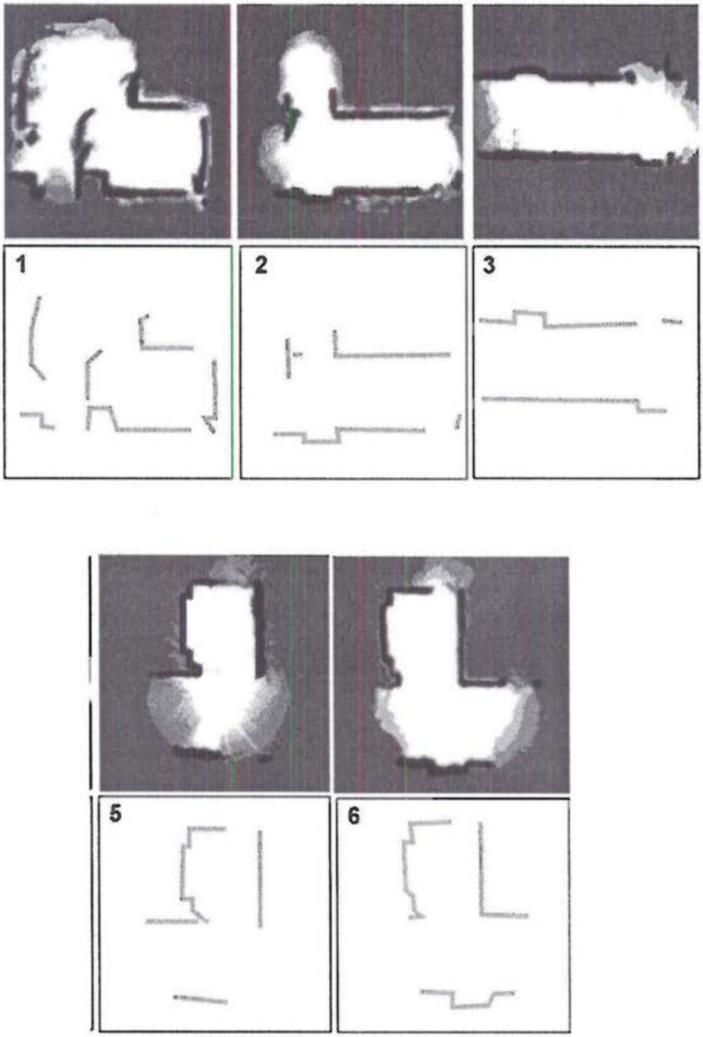

Figure 2. Local grid and vector maps built from laser and sonar data

An example of the sensory-data fusion and of the conversion between representations in the MAB framework is shown in Fig. 2.

In this experiment, the robot followed a path in the hallway and the lab-

SOURCE	**EVENT**	**TRIGGERED AGENT**	**ACTION**
ADCtrl	CtrlSt modification	AEReport	analyze the controller status
	SelfLoc setting	ADSonar	suspend the action
		ADScan	build the LocalVectorMap
		AEMapConv	build the VectorMapFromGrid
ADSonar	GridMap modification	AEPilot	detect obstacles
	SonarSt modification	AEReport	analyze the sonar status
ADScan	GridMap modification	AEPilot	detect obstacles
	LocalVectorMap creation	AEMapFuz	build the CurrentVectorMap
	ScanSt modification	AEReport	analyze the scanner status
ADComm	InMsg modification	AEPosUpd	evaluate the CurrPos
		AEMapUpd	update the GlobalVectorMap
		AEReport	analyze messages
ADCam	CamPos modification	AEPosUpd	detect obstacles
	CamSt modification	AEReport	analyze the camera status
AEMapConv	VectorMapfromGrid creation	AEMapFuz	build the CurrentVectorMap
AEMapFuz	CurrentVectorMap creation	AEPosSens	evaluate the SensPos
AEMapUpd	OutMsg modification	ADComm	send messages
AEPilot	NextMove setting	ADCtrl	send command to controller
AEReport	OutMsg modification	ADComm	send messages
AEPosUpd	CamLoc setting	ADCam	evaluate the CamPos
	OutMsg modification	ADComm	send messages
	CurrPos modification	AEMapUpd	update the GlobalVectorMap
		ADCtrl	resume the action
		ADSonar	resume the action
		ADScan	resume the action
AEPosSens	SensPos modification	AEPosUpd	evaluate the CurrPos

Table 1. The event-action dependencies

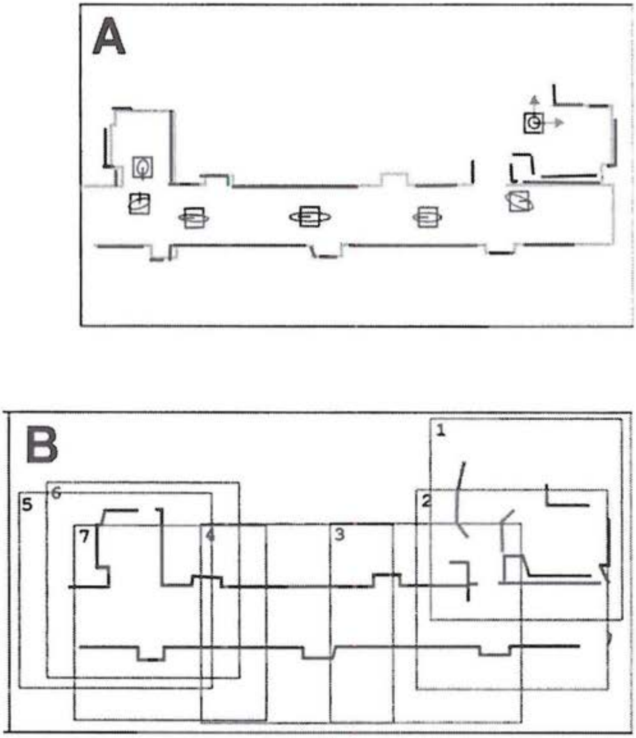

Figure 3. Laser-based line segments (A) and the global vector map (B) - rectangles show the overlapping local maps

using scanner and sonar data, and the parametric models extracted from the respective grids by the *AEMapConv* agent, using the Hough transform.

In Fig. 3A the line segments generated by the laser scanner agent *ADScan* are shown, overlaid on the predefined environment model used for self-localization. The line segments obtained from the laser scanner have also contributed to the global vector-based map. It can be seen from this figure that the scanner did not recognize some objects in the environment, e.g. the tentative wall in the lab, made of a few styrofoam boards - this object is below the plane of the laser beam. The ellipses shown on robot icons represent the position uncertainty at these points, Dudek et al. (2000). Note that the uncertainty of the position in the corridor was quite large due to the lack of perpendicular into account the parametric interpretations of the multi-sensor local maps. It is much more complete than the vector map obtained from the unimodal sensory data. This is attributed to the delayed feature recognition. The line segments are retrieved from the grid representation after several (about 10 in this experiment) consecutive map updating cycles, thus, the system avoids solving the explicit data association problem on the basis of a single measurement, or a set of measurements taken from a single vantage point. However, some inconsistencies persist in the map, being the result of the limited self-localization accuracy - the local maps have been registered within the global frame using the position estimates shown in Fig. 3A.

6. Final remarks

The paper describes the multi-agent blackboard (MAB) architecture of the mobile robot, which is aimed at the data-driven sensor information processing resulting in the representation of the robot environment (collection of maps). The structure of the blackboard, the set of agents and their tasks are presented. The control aspects of architecture together with the sources of events and the event handling actions are also presented. This architecture guarantees a flexible use of sensors and world (environment) representations.

The current research concerns the reasoning about the environment (Brzykcy at al., 2001c).

References

ARKIN, R. (1998) *Behavior-Based Robotics.* The MIT Press, Cambridge.

BIGUS, J.P. and BIGUS, J. (1998) *Constructing Intelligent Agents with Java. A Programmer's Guide to Smarter Applications.* John Wiley & Sons.

BORENSTEIN, J. and KOREN, Y. (1991) Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance. *IEEE Journal of Robotics and Automation* **7** (4), 535-539.

BRZYKCY, G. (2000) Koncepcja wieloagentowego systemu percepcji i modelowania otoczenia (A concept for multi-agent system of perception and modelling of the environment; in Polish). In: Bubnicki Z. and Grzech A., eds., *Inżynieria wiedzy i systemy ekspertowe.* Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, **2**, 111-118.

BRZYKCY G., MARTINEK J., MEISSNER A. and SKRZYPCZYŃSKI P. (2001a) Blackboard Agent Architecture for Perception and World Modeling. *Automation 2001*, Warszawa, 212-219.

BRZYKCY G., MARTINEK J., MEISSNER A. and SKRZYPCZYŃSKI P. (2001b) Multi-Agent Blackboard Architecture for a Mobile Robot. *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, Maui, 2369-2374.

BRZYKCY, G., MARTINEK, J., MEISSNER, A. and SKRZYPCZYŃSKI, P. (2001c) System wnioskowania robota o scenie z prostymi obiektami geometrycznymi (A system of reasoning of a robot on a scene with simple geometric objects; in Polish). *III Konferencja "Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim"*, Kraków, 193-198.

DUDEK, G. and JENKIN, M. (2000) *Computational Principles of Mobile Robotics.* Cambridge Univ. Press.

ENGELMORE, R. and MORGAN, T. (eds.) (1988) *Blackboard Systems.* Addison-Wesley.

HAYES-ROTH, B. (1985) A Blackboard Model for Control. *Artificial Intelligence* **26**, 251-322.

HUNHNS, M. and SINGH, M. (eds.) (1998) *Readings in Agents.* Morgan Kaufmann Publishers.

KAPPEY, D., POKRANDT, P. and SCHLOEN, J. (1994) A distributed multi-sensor blackboard system for autonomous robot. *SPIE Conference Sensor Fusion VII*, Boston.

KASIŃSKI, A. and BACZYK, R. (2001) Robust Landmark Recognition with Application to Navigation. *Proc. Conf. Computer Recognition Systems (KOSYR)*, Wrocław, 401-407.

KASIŃSKI, A. and SKRZYPCZYŃSKI, P. (1998) Cooperative Perception and World-Model Maintenance in Mobil Navigation Tasks. In: Lueth T. et al., eds., *Distributed Autonomous Robotic Systems 3.* Springer, Berlin, 173-182.

KASIŃSKI, A. and SKRZYPCZYŃSKI, P. (2002) Communication Mechanism in a Distributed System of Mobile Robots. In: Asama H. et al., eds., *Distributed Autonomous Robotic Systems 5.* Springer-Verlag, Tokyo, 51-60.

MÜLLER, J. (1995) *The Design of Intelligent Agents: a Layered Approach.* Lecture Notes in Artificial Intelligence, 1177, Springer.

SCHWARTZ, D. (1995) *Cooperating heterogeneous systems.* Kluwer Academic Publishers.

SKRZYPCZYŃSKI, P. and DRAPIKOWSKI, P. (1999) Environment Modelling for a Multi-Agent Mobile System. *Proc. EUROBOT'99*, Zurich, 41-48.