DECENTRALIZED CONTROL OF DISTRIBUTED INTELLIGENT ROBOTS AND SUBSYSTEMS

TH. LAENGLE and T.C. LUETH

University of Karlsruhe, Faculty for Informatics, Institute for Real-Time Computer Systems and Robotics, Prof. Dr.-Ing. U. Rembold, Prof. Dr.- Ing. R. Dillmann, 76128 Karlsruhe, Germany

Abstract. In a distributed robot system, asynchronous and synchronous communication between the system components is necessary to guarantee problem soliving capability in realtime. On that account, the distributed control architecure of the Karlsruhe Autonomous Mobile Robot KAMRO which is being developed at IPR has been extended by these communication kinds. The robot system consists of several subcomponents, like manipulators, two hand-eyecameras, one overhead camera and a mobile platform. To get better problem solving capability than the former centralized control architecture, these components are able to work together in teams (asynchronous communication) or special agents (synchronous communication).

Key Words. Distributed control, real-time computer systems, robots, control theory, robot control architecture

1. INTRODUCTION

Throughout the years, the trend has clearly gone toward large and more complex systems that are used for the intelligent resolution of problems in the area of industrial production and maintenance. Autonomously guided vehicles and autonomous manipulator systems that were constructed for service tasks are examples (Rembold *et al.*, 1993; Schraft, 1993). An example for an autonomous service system is the mobile robot KAMRO (see Fig. 1) developed at IPR (Lueth and Rembold, 1994).



Fig. 1: The autonomous assembly robot KAMRO

The main problem is the construction of the control architecture for such complex systems. This architecture is also often termed intelligent architecture for planning and control (IAPC). In principle, complex systems that consist of several executive subsystems (agents) can be divided into three different design classes (Lueth and Laengle, 1994b). The same distinction can be made for planning systems (Hayes-Roth, 1985; Thorpe, 1990; Arkin, 1989):

- *Centralized Systems:* A decision is made in a central mechanism and transmitted to executive components (see Fig. 2a).
- *Distributed Systems:* The decision is made by a negotiation process between the executive components and executed by them (see Fig. 2b).
- *Decentralized Systems:* Each executive component makes its own decisions and executes only these decisions (see Fig. 2c).



Fig. 2: Execution view of multi-agent systems

Centralized control architectures are often divided into functional-based (Albus, 1991) and behaviourbased (Brooks, 1986) systems. In both architectures, the number and complexity of the hierarchical system levels are responsible for reaction time and for task execution itself.

The development of these architectures is very difficult because it's not easy to determine the suitability of this concept in advance. It is also often required to integrate new sensors or actuators to the system, or the system may need an on-line cooperation of system components. This integration process is very problematic due to the complexity of the hierarchical planning system. When integrating new components into an already existing system, the disadvantages overwhelm what was previously thought of as the advantage of the simpler original construction of those centralized system control architectures. This is because the world representation and the control architecture must be refined, or new procedures for cooperation between system components must be implemented. In very difficult cases, the control architecture must be reimplemented to obtain an integration on the same or a higher level. The satisfaction of a realtime requirement can only be tested and verified online. Furthermore, fault-tolerance and error recovery is hard to obtain (Trevelyan and Nelson, 1987; Srinivas, 1977; Hörmann and Rembold, 1991).

In contrast to that, distributed or decentralized control architectures show their main advantages, when it's necessary to enhance the system, to integrate components, and to maintain the system. This new concept consists of local intelligence, local world representation, decentralized communication, dynamic teams and uses concepts of distributed artificial intelligence (Bond and Gasser, 1988). New is the transfer of these concepts to a complex autonomous task execution (like assembly), the consideration of different communication channels, the guarantee of faulttolerance and error recovery in a distributed controlled robot system (Lueth and Laengle, 1994a) and the solution of hand-eye or eye-hand control problems in real-time (Kawauchi et al., 1993; Habib et al., 1992, Yuta and Premvuti, 1992).

In the second section, the distributed control architecture KAMARA for the mobile assembly robot KAMRO is shortly described and the necessity for asynchronous (teams, section three) and synchronous (special agent, section four) comunication between the system components is proved. In the fifth section, extensions to the distributed robot control architecture KAMARA are explained that are responsible for team building and special agents. Furthermore, it deals with the assessment of the real-time aspects. The article ends with an evaluation and conclusion for future work.

2. THE ARCHITECTURE KAMARA

The formerly centralized control architecture FATE (Hörmann *et al.*, 1991) consists of a blackboard planning level that generates situation-dependent manipulator-specific elementary operations, and a real-time robot controller RT-RCS that executes the elementary operations. The real-time controller is able to control the manipulators independently or in a closed cinematic chain.

In the KAMARA concept that is described briefly in (Lueth and Laengle, 1994b), the responsibility of coordination and scheduling is distributed to all system components. This distribution should be homogeneous over all system components making it possible to be assisted by other systems or subsystems. As a consequence, a recursive algorithm is used to exchange tasks on the cell-level (cooperation of different robot systems) and on the robot level (cooperation of the components of a robot, for example, cameras, manipulators, platform) (see Fig. 3) and can therefore be used as an integrated system.

If a system component requires information for problem solving which it cannot, or just insufficiently, calculate by itself, then it can relate to other system components to receive a problem solution - independent to the system level (robotlevel or cell-level) these components belong.



Fig. 3: Decentralized system on cell level (MAS) and robot level (DRS)

By the introduction of an agent definition, it is possible to describe and explain hierarchical systems (see Fig. 4). An agent consists of three parts. The *communicator* connects the head to other agents on the same or on higher communication levels. The *head* is responsible for the action selection process for the body. The *body* itself consists of one or more executive components, which can be considered as agents in the same way.



Fig. 4: Elements of an agent

In principle, this multi-agent architecture is also useful on the cell level. In this case, the communication mechanism of one KAMRO is the head of a KAMRO agent and it is possible to use more than one KAMRO for complex tasks, like carrying a large object with several robots or loaning one manipulator to a second robot (Fig. 5).



Fig. 5: Distributed cell controller view

In KAMARA, it's important to have the same cooperation capabilities, for example, closed cinematic chains for coupled agents, as in the centralized system: synchronous and asynchronous communication. Main topic of the following sections will be the description of these cooperation types.

3. ASYNCHRONOUS COMMUNICATION

Considering the distributed control architecture, it's easy to notice that on many accounts the agents have to build teams to solve specific tasks. These teams are dynamic in nature, the number as well as the kind of agents may change during the task execution. An example is the exchange of parts between both manipulators. For a defined space of time, cooperation is necessary to reach this goal. The communication for this kind of cooperation will be done on a high abstraction level by the agent's communicator.

As mentioned before, an agent A consists of a communicator, a head and a body. In this system description, an agent, like a manipulator, is only capable to perform one task at a time. That is why its body B is implemented as a single procedure. On the other side, the head with the communicator must not only control the body, but also has to communicate and negotiate with other agents or heads. Communication is important to start and to execute the decision making process to determine the agent for executing an elementary operation. This means, the head (and same the communicator) has to deal with several different tasks at one time. Therefore, head and communicator are implemented as a variable set H, C of equal independent processes H, C for planning, communication, and negotiation (see Fig. 6):

$$A = (\mathbf{C}, \mathbf{H}, B) \tag{1}$$



Fig. 6: Head and Communicator can be several processes

As an example, a team of the components camera and manipulator of the autonomous mobile assembly robot KAMRO can be considered. The cooperation of manipulator and camera system is important if grasping and joining operations should be performed by the robot. In this case, the camera system must be able to correct the position of the manipulator if needed. Another example is the exchange of an obstacle between both manipulators, or a regrasping operation to change the gripping configuration of an object. In the first case, the camera and a manipulator must build a team to solve the described problem, in the second case, both manipulators together build a team. Because there is just brief information exchange that has not to be synchronized in a specific time interval, team building (see Fig. 7) is sufficient.



Fig. 7: Team: asynchronous communication

The communication form between the system agents is asynchronous. On that account, it is not possible to guarantee real-time constraints. If realtime constraints are important, it's necessary to use a special agent as described in the next section.

4. SYNCHRONOUS COMMUNICATION

If two manipulators grasp a large or heavy part, and by this way close a cinematic chain, asynchronous communication between these system components is not sufficient. In this case, dependent on the desired control concept for the cinematic chain, a decentralized architecture (simple reflexive behaviour), a distributed architecture (master slave tasks), or a centralized architecture is required. In some cases (for example, complex two-arm manipulation tasks), a centralized robot controller is better than any other approach at this moment. This is the reason for an extension of the distributed control concept by the introduction of special executive agents. These special agents SA have, like all other agents, a head H and a communicator C. The body is allowed to allocate bodies of other agents, if available, and control them by special communication channels with high transfer rates. During this time, the normal agents have no access to their bodies, since they are used by the special agent (see Fig. 8). Because special agents change the structure of the control architecture while they are active, they should only be used if no other type of cooperation is suitable.

In other words: if the information exchange between agents of the same team increases so dramatically that this results in a narrowing in the communication channels, then these agents have the possibilities to refer to a closer internal relationship (see Fig. 9).



Fig. 8: Special Agent: Centralized planning for other agent bodies

This way, this special agent acts as a single component in the system. This is the case when two manipulators transport a single obstacle: a team of two systems, perhaps two robots, is not possible due to communication speed, but a special agent can solve this problem due to its faster communication capability.



Fig. 9: Communication for high transfer rate

5. REAL-TIME CONSTRAINTS

As described in the last two sections, teams and special agents are necessary in distributed robot control systems to guarantee better problem-solving capabilities than centralized systems. The communication form in special agents is synchronous, whereas the communication in teams is asynchronous (and not so fast). In the KAMARA system, task execution planning is performed by the agent head. An assembly task KAMARA has to perform is therefore represented by a precedence graph whose nodes consist of individual subtasks. As an example, the Cranfield Benchmark shown in Fig. 10 is considered.



Fig. 10: The Cranfield Assembly Benchmark

The precedence graph (see Fig. 11) only describes the goals the system has to reach whereas the executing agent has to decide how these goals can be achieved depending on the environment at execution time. At execution time, the agent head uses the system's sensor information to expand this implicit representation to an explicit one that can be executed by the agent body.



Fig. 11: Assembly Precedence Graph

The new control architecture KAMARA uses several components of the former centralized robot architecure. This way, the execution of an explicit elementary operation is in both systems performed by the real-time robot control system RT-RCS. RT-RCS is capable to execute these operations in real-time. On that account, the bottle-neck of the KAMARA-system is the communication between the system components, as a consequence, the communication kinds have to be examined in detail.

To investigate the real-time constraints briefly, both manipulators of our robot system KAMRO are

considered. In the KAMARA system, two manipulator agents exist that are able to perform the assembly task:

Manipulator: A single manipulator is able to perform the implicit elementary operations PICK and PLACE if the pick/place-position is in the same working area.

Two-arm-manipulator: A two-arm-manipulator is also able to perform the implicit elementary operations PICK and PLACE. Because this agent consists of two independent actors that build a special agent as described above, the mission valuation of this agent (in the negotiation process) is much higher than the calculated value of a single manipulator when the system must PICK a heavy or great obstacle.

A *team* of both manipulators is not modeled by the use of a seperate agent, because team-building is performed on-line if the robot has to regrasp or exchange an obstacle.

While execution of an implicit elementary operation, four different situations may occur:

- One manipulator has the ability to PICK and PLACE the obstacle. As a consequence, there is no need for a manipulator team or special agent. Real-time constraints are met because only one manipulator is needed, and the real-time robot control system RT-RCS that has to execute the operations guarantees these contraints.
- 2. One manipulator has the ability to PICK the obstacle, the other manipulator must PLACE it. In this case, both manipulators build a team, no real-time constraints can be guaranteed because the information exchange is asynchronous. If real-time constraints are important, it is possible to introduce a new exchange-special-agent that performes exchange-missions if, for example, a real-time flag is set in the mission descripion. This way, both manipulators are controlled by the use of one planning component, and the communication form is no longer asynchronous. In all other cases, team-building is sufficient.
- 3. The manipulator has the ability to pick the obstacle, but it is not possible to place or join it because the gripping configuration is not right. In these cases, a regrasping operation is necessary to change the gripping configuration, and the other manipulator is used to hold the object. As a consequence, both manipulators build a team, It is possible to solve the problem similar to the exchange-operation.
- 4. The obstacle must be picked by both manipulators. Therefore, they have to build a

special agent. Because the decision making is done in a centralized way in this case, and the two arm operations are executed by RT-RCS, real-time constraints can be guaranteed.

Thus, it is possible to meet real-time contraints in either case of manipulation. Problems are raising only if the communication form on the planning level is asynchronous, in these cases time stamps can be used to request other agents to react in short time, or new special agents are introduced.

6. CONCLUSION

In this article, a new architecture for intelligent technical system control has been presented. The described concept uses local intelligence, decentralized communication, teams and special agents. As a testbed, the autonomous assembly robot KAMRO is used. Future work should verify the advantages of the distributed robot control architecture. This way, it should also be possible to integrate the mobile platform in the assembly process to assist the manipulators.

7. ACKNOWLEDGEMENT

This research work was performed at the Institute for Real-Time Computer Systems and Robotics (IPR), Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, University of Karlsruhe and is funded by the national basic research project on artificial intelligence (SFB 314) funded by the German Research Foundation (DFG).

8. REFERENCES

- Albus, J.S. (1991). Outline for a Theory of Intelligence. *IEEE Trans. on System Man and Cybernetics*, Vol. 21.3, pp. 473-509
- Arkin, R.C. (1989). Motor Schema-Based Mobile Robot Navigation. *The International Journal of Robotics Research*, Vol. 8.4, pp. 92-112.
- Bond, A. H., and L. Gasser (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers.
- Brooks, R.A. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Trans. on Robotics and Automation*, Vol. 2, pp. 14-23.
- Habib, M.K., Asama, H., Ishida, Y., Matsumoto, A., and I. Endo (1992). Simulation Environment for an Autonomous and Decentralized Multi-Agent Robotic System. *IROS IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 7-10, pp. 1550-1557.

- Hayes-Roth (1985). A blackboard architecture for control. *Artificial Intelligence*, Vol. 26, pp. 251-321.
- Hörmann, A., Meier, W., and J. Schloen (1991). A control Architecture for an advanced fault-tolerant robot system. *Robotics and Autonomous Systems*, Sacramento, California, Vol. 7, pp. 211-225.
- Hörmann, A., and U. Rembold (1991). Development of an Advanced Robot for Autonomous Assembly. *ICRA IEEE Int. Conf.* on Robotics and Automation, pp. 2452-2457.
- Kawauchi, Y., Inaba, M., and T. Fukuda (1993). A Principle of Distributed Decision Making of Cellular Robotic System (CEBOT). *ICRA IEEE Int. Conf. on Robotics and Automation*, Atlanta, pp. 833-838.
- Lueth, T.C., and U. Rembold (1994). Extensive Manipulation Capabilities and Reliable Behaviour at Autonomous Robot Assembly. Accepted at: *IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, May 8-13
- Lueth, T.C., and T. Laengle (1994a). Fault-Tolerance and Error Recovery in an Autonomous Robot with Distributed Controlled Components. Accepted at: *Int. Symposium on Distributed Autonomous Robotic Systems*, Riken, Japan, July 12-15, 1994
- Lueth, T.C., and T. Laengle (1994b). Task Description, Decomposition, and Allocation in a Distributed Autonomous Multi-Agent Robot System. Accepted at: *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Munich, Germany, Sept 12-16
- Rembold, U., Lueth, T., and A. Hörmann (1993). Advancement of Intelligent Machines. ICAM JSME Int'l Conf. on Advanced Mechatronics, Tokyo, Japan, August 2-4, pp. 1-7.
- Schraft, R.D. (1993). Vom Industrieroboter zum Serviceroboter - Stand der Technik, Chancen und Grenzen. *ISRR VDI/VDE-GMA Intelligente Steuerung und Regelung von Robotern*, pp. 1ff
- Srinivas, S (1977). Error recovery in robot systems. PhD Theses, California Institute of Technology.
- Thorpe, C.E. (1990). Ed. Vision and Navigation -The Carnegie Mellon Navlab, Kluwer Academic Publishers, Boston.
- Trevelyan, J.P, and M. Nelson (1987). Adaptive robot control incorporating automatic error recovery. *International Conference on Advanced Robotics.*
- Yuta, S., and S. Premvuti (1992). Coordinating Autonomous and Centralized Decision Making to Achieve Cooperative Behaviours between multiple mobile robots. *DARS Int. Symp. on Distributed Autonomous Robotic System*, Wako, Japan, Sep. 21-22, pp. 173-181.