9th Intern. Symp. on Experimental Robotics 2004 (ISER 04), Singapore, 18-21 June 2004

# Coordinated Task Execution for Humanoid Robots

T. Asfour, D.N. Ly, K. Regenstein, and R. Dillmann

Forschungszentrum Informatik Karlsruhe (FZI)
Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany
http://www.fzi.de
Email: asfour@ira.uka.de, ly@fzi.de, regenstein@fzi.de, dillmann@ira.uka.de

**Abstract.** This paper presents a framework for the coordinated execution of tasks in robotic systems with a high degree of freedom such as humanoid robots. Focusing on tasks to be executed by different subsystems of the robot (e.g. mobile platform, two redundant arms, a head with vision and acoustic system), a motion coordination scheme is presented. The coordination scheme is based on the synchronization of the motion of each subsystem while performing a common task. The validity of the proposed coordination scheme is experimentally demonstrated by different tasks of the humanoid robot e.g. two-arm tasks, head-arm tasks or platform-arm tasks.

## 1 Introduction

Our current research interest is the development of a control architecture to achieve manipulation task goals for a humanoid robot. In particular, we address the programming and coordinated execution of manipulation tasks in a household environment. Therefore, it is an important issue to coordinate the multiple subsystems of a humanoid robot in carrying out tasks in dynamic unstructured environments. The control architecture must provide the possibility to integrate the motor, perception and cognition components necessary for natural multimodal human-humanoid interaction and human-humanoid cooperation. In the literature, considerable research effort has been focused on various problems related to motion coordination of robot systems. So far,many coordination schemes for multiple arm systems have been reported: the master/slave control [7], the centralized control [14] and the decentralized control ([8], [6]). To design control architectures for humanoid robots, some approaches have been suggested: subsumption architecture [4] and task-oriented approaches [10]. This paper is organized as follows. Section 2 briefly describes the humanoid robot ARMAR. The control architecture for coordinated task execution is introduced in section 3. The coordination strategy using Petri nets is given in Section 4. Section 5 presents the implementation and the experimental results of coordinated execution of different tasks.

## 2 System Configuration – The Humanoid Robot ARMAR

The humanoid robot ARMAR [3] has 23 mechanical degrees-of-freedom (DOF). From the kinematics control point of view, the robot consists of five subsystems:

Head, left arm, right arm, torso and a mobile platform. The upper body of ARMAR has been designed to be modular and light-weight while retaining similar size and proportion as an average person. The control system of the robot is divided into separate modules. Each arm as well as torso, head and mobile platform having its own software- and hardware controller module. The head has 2 DOFs arranged as pan and tilt and is equipped with a stereo camera system and a stereo microphone system. Each of the arms has 7 DOFs and is equipped with 6 DOFs force torque sensors (ATI, [15]) on the wrist. The current mobile platform of ARMAR consists of a differential wheel pair and two passive supporting wheels. It is equipped with front and rear laser scanner (Sick, [17]). Furthermore, it hosts the power supply and the main part of the computer network.


**Fig. 1.** The humanoid robot ARMAR with five subsystems: head, two arms, torso and mobile platform.

## 3 Control Architecture for Humanoid Robots

In this section, we introduce our control architecture. First, we summarize the design criteria. Second, we introduce the proposed and hierarchically organized control architecture. The control architecture was designed according to the following global criteria:

- Flexibility and modularity to cope with various tasks and to allow the addition of further tasks and hardware and software modules in a simple manner. This is a very important feature for the process of integration.
- Real-time performance to allow a prompt response to varying environments and exceptions which can occur during the task execution.

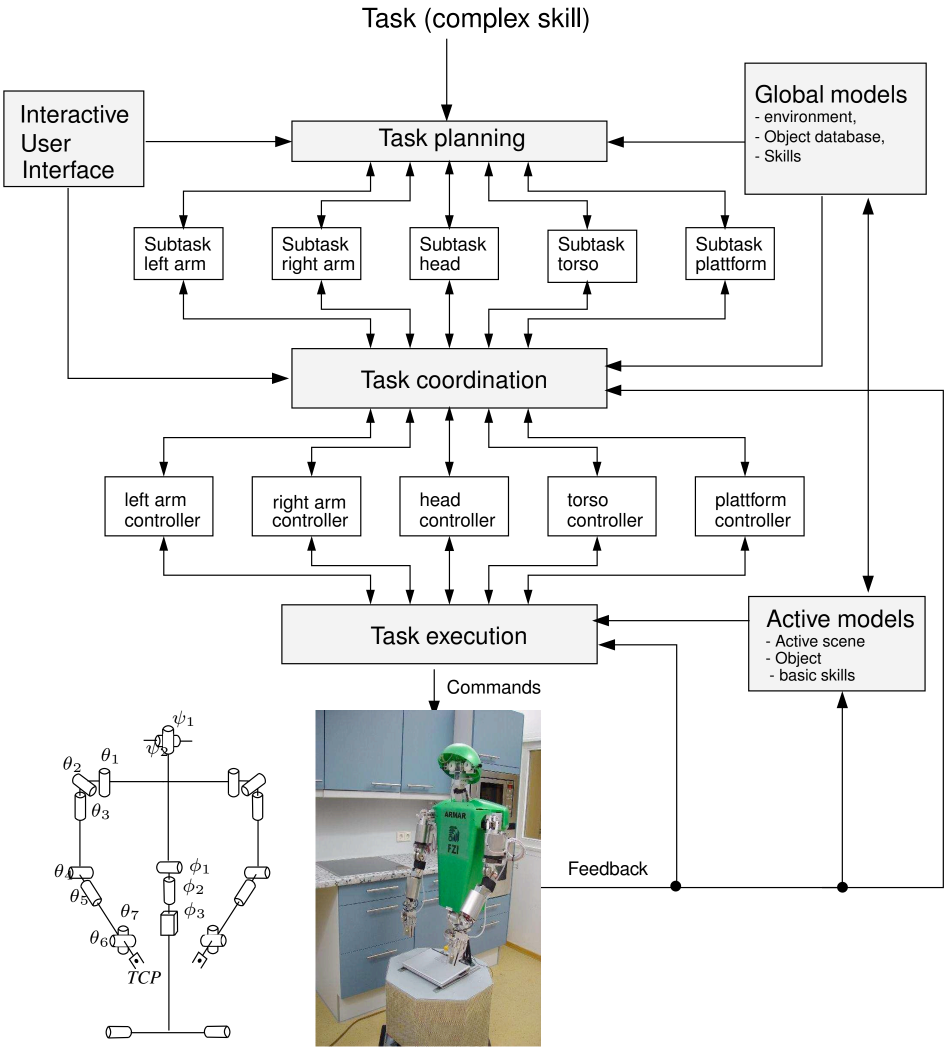



**Fig. 2.** Hierarchical control architecture for coordinated task execution in humanoid robots: planning, coordination and execution level.

According to the definition of intelligent machines given by Sardis ([12], [13]) we decomposed the overall control system of the humanoid Robot ARMAR into three levels as shown in figure 2. A given task is decomposed into several subtasks, representing the sequence of actions the subsystems of the humanoid robot must carry out to accomplish the task goal. The coordinated execution of a task requires the scheduling of the subtasks and their synchronization with logical conditions, external and internal events. Figure 2 shows the block diagram of the control architecture with three levels:

- The task planning level specifies the subtasks for the multiple subsystems of the robot. This level represents the highest level with functions of task representation. It generates the subtasks for the different subsystems of the robot autonomously or interactively by a human operator. The generated subtasks for the lower level contain the whole information necessary for the task execution, e.g., parameters of objects to be manipulated in the task or the 3D information about the environment. According to the task description, the subsystems controllers are selected here and activated to achieve the given task goal.
- The task coordination level generates sequential/parallel primitive actions for the execution level in order to achieve the given task goal. The subtasks are provided by the task planning level. Like on the planning level the execution of the subtasks in an appropriate schedule can be modified/reorganized by an operator using an interactive user interface.
- The task execution level is characterized by control theory to execute specified sensory-motor control commands. This level uses task specific local models of the environment and objects. In the following we refer to those models as *active models*.

The active models are first initialized by the global models and can be modified and enhanced during the progress of the task execution. Internal system events and execution errors are detected from local sensor data. These events/errors are used as feedback for the task coordination level in order to take appropriate measures. For example, a new alternative execution plan can be generated to react to internal events of the robot subsystems or to environmental stimuli.

## 4 Coordination Strategy for Task Execution

The execution of coordinated tasks demands a mechanism for synchronization of actions allowing a deterministic switch between sequential/parallel actions of the robot. Therefore, a framework for coordinated execution of tasks using condition/event Petri nets was developed. Among the existing models of discrete event systems, Petri nets have been widely used to model dynamic systems [5]. In our work, we use Petri nets to efficiently represent both control and data flow within one formalism.

### 4.1 Petri nets

Petri nets have been widely used in both theoretic works to model dynamic systems and applications, especially in the modeling of manufacturing processes Petri nets are a graphical and mathematical formalism for modeling, simulation and formal analysis of discrete event systems [5], [1]. Petri nets allow the representation of both control and data flow within one formalism. There are many Petri net variants and definitions, which are extentions of so-called condition/event nets:

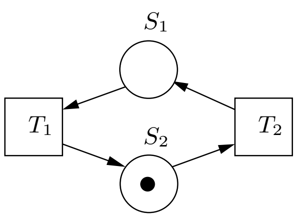
**Definition (Condition/event net)** A condition/event Petri net is defined by the 4-tuple $N = (P, T, A, m_0)$, where

- $P = \{p_1, \ldots, p_{n_p}\}$ is a finite set of places,
- $T = \{t_1, \ldots, t_{n_t}\}$ is a finite set of transitions,
- $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $m_0 : P \rightarrow \{0, 1\}$ is the initial marking. It defines the initial number of undistinguishable tokens on each place $p \in P$.
- **Enabling rule:** A transition $t_j \in T$ is enabled if all input places of $t_j$ contain a token and all output places are empty.
- **Firing rule** An enabled transition may fire. On firing it removes the tokens from all its input places and places one token in each of its output places.

The set of places describes the states of the system, and the set of transitions defines events that can change the state of the system. The state of a Petri net is represented by its markings, i.e., the distribution of tokens among places. In the usual graphical representation of Petri net graphs, places are depicted as circles and transitions as rectangles. The marking, that is, the distribution of tokens on places, represents the state of a Petri net model. Transition firings change the token distribution and thus the state of the system. They may reflect the occurrence of events or the execution of an operation.

### 4.2 Coordination of Task Execution

Figure 3 represents a Petri net for modeling one subsystem with the associated places and transitions. The shown initial marking indicates the state *ready* of the arm. The task execution can be invoked by firing the transition $T_2$ which leads to the state *active*.




| Conditions (places) | |
|---|---|
| $S_1$ | Subsystem is active |
| $S_2$ | Subsystem is ready |

| Events (Transitions) | |
|---|---|
| $T_1$ | Task execution is completed |
| $T_2$ | Task execution is not completed |

**Fig. 3.** Petri net for modeling of one subsystem (left) with its conditions and events (right).

The Petri net in figure 4 results from the synchronous composition of three nets for the coordinated execution of three tasks through the three different subsystems $A$, $B$ and $C$. The descriptions of the transitions and events including the pre- and postconditions are given in table 1 and in table 2. The coordination of different subsystems motion takes place through the common transitions 7, 8, 9 and 10. Firing

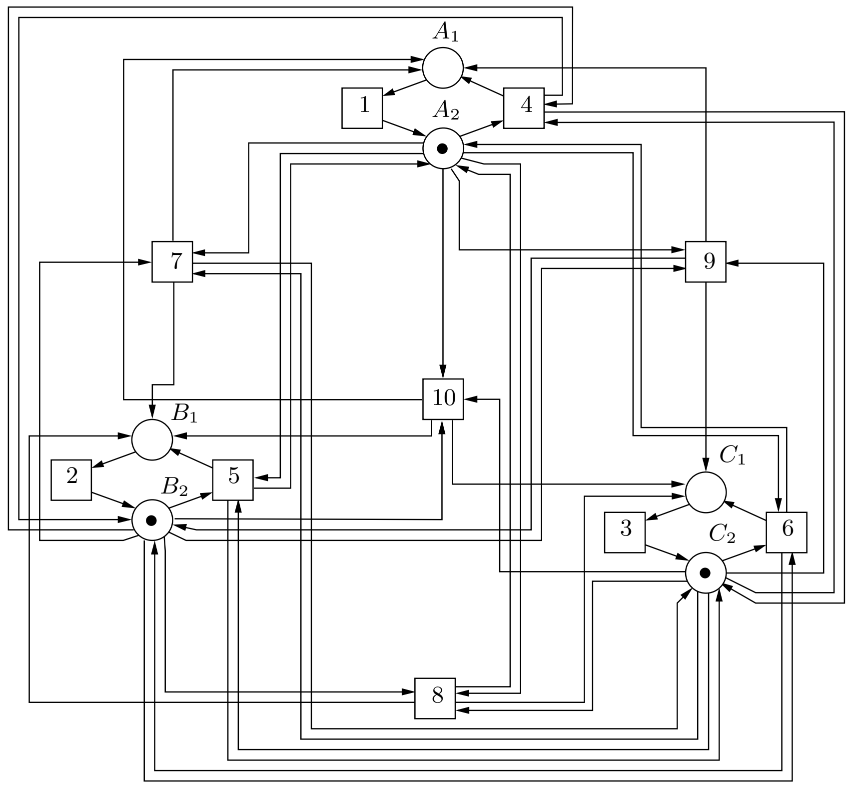



**Fig. 4.** Petri Net for coordinated task execution of three subsystems $A$, $B$ and $C$

the transition 7 denotes respectively the task execution through the subsystems $A$ and $B$, under the condition that both subsystems are ready. Firing of transition 10 denotes the simultaneous and parallel execution of tasks through the subsystems $A$, $B$ and $C$.

## 5 Implementation and Experiments

The different coordination nets for multiple cooperative tasks of the robot subsystems are modeled and implemented in the software control framework MCA ([16]). It provides a standardized module concept with unified interfaces and allows an easy building of control groups with more complex functionalities from basic modules. Furthermore, it provides graphical tools for debugging and visualization of the groups structure mentioned above as well as a graphical user interface with various entities to input motion commands and to output the state of sensor values, parameters or return values at the different levels of the architecture. The control architecture is also mapped into a hardware control system. At the task planning and coordination

| Conditions | Description |
|---|---|
| $A_1$ | Subsystem $A$ is active |
| $A_2$ | Subsystem $A$ is ready |
| $B_1$ | Subsystem $B$ is active |
| $B_2$ | Subsystem $B$ is ready |
| $C_1$ | Subsystem $C$ is active |
| $C_2$ | Subsystem $C$ is ready |

**Table 1.** Conditions for the Petri net in figure 4

| Events | Description | Pre-condition | Post-condition |
|---|---|---|---|
| 1 | Task of subsystems $A$ is completed | $A_1$ | $A_2$ |
| 2 | Task of subsystems $B$ is completed | $B_1$ | $B_2$ |
| 3 | Task of subsystems $C$ is completed | $C_1$ | $C_2$ |
| 4 | New task for subsystem $A$ | $A_2$, $B_2$, $C_2$ | $A_1$, $B_2$, $C_2$ |
| 5 | New task for subsystem $B$ | $A_2$, $B_2$, $C_2$ | $A_2$, $B_1$, $C_2$ |
| 6 | New task for subsystem $C$ | $A_2$, $B_2$, $C_2$ | $A_2$, $B_2$, $C_1$ |
| 7 | New tasks for subsystems $A$ and $B$ | $A_2$, $B_2$, $C_2$ | $A_1$, $B_1$, $C_2$ |
| 8 | New tasks for subsystems $B$ and $C$ | $A_2$, $B_2$, $C_2$ | $A_2$, $B_1$, $C_1$ |
| 9 | New tasks for subsystems $A$ and $C$ | $A_2$, $B_2$, $C_2$ | $A_1$, $B_2$, $C_1$ |
| 10 | New tasks for subsystems $A$, $B$ and $C$ | $A_2$, $B_2$, $C_2$ | $A_1$, $B_1$, $C_1$ |

**Table 2.** Transitions and their meaning in the Petri net in figure 4

levels embedded PCs with broadband bus system (firewire) are used whereas DSP-FPGA-based controller modules are used at the execution level. They communicate with the embedded PCs via CAN-Bus. All components of the embedded control components are programmed and running under Linux, kernel 2.4.20 with the Real Time Application Interface RTAI 24.1.11. The coordination strategy has been applied and implemented in our humanoid robot. Several manipulation and locomotion experiments have been carried out to prove its suitability.

**Mobile manipulation tasks:** Mobile manipulation capabilities are the key of humanoid robot applications in household environments. The navigation and manipulation methods are integrated to execute service tasks like taking a bottle from a person, collision free driving to a mission target and placing it onto a table. In the case of this scenario the firing of the transitions of the associated coordination nets is specified by different events. At the beginning all subsystems of the robot arm are in the initial state *ready*. One of the arms, e.g., the right arm switches to the state *active* when the person is detected and starts the execution of the grasping task. In the case of a successful grasp, the mobile platform starts driving to the target position. The placing operation is then done through coordinated actions of the controllers of the right arm, platform and head. In all phases of the task each motion controller periodically acquires local sensory data and interprets it in order to take suitable measures.

**Dual arm tasks:** As an initial investigation of methods for dual arm cooperative tasks, we chose to implement an object level controller for the control of the position of the center of a lightweight dumbbell, which has to be lifted up by both arms. The arm trajectories are derived from the object trajectory. In order to achieve a coordinated execution of the task, synchronization points are included along each arm trajectory. These points represent conditions for firing of associated transitions in the coordination net.

**Visual person tracking and auditory tracking:** In order to make the interaction with the robot easier and more reliable, we integrated the algorithms, which has been originally developed in [11] for the visual perception of the user into the control system of our robot to demonstrate the motion coordination of the head and the mobile platform. The 3D-positions of the user's head and hands are mapped into joint angles of the robot head and arms. For the mapping of the hand positions into a robot arm postures, we use the method presented in [2] in order to generate human-like arm postures. Furthermore, an acoustic localization algorithm using a stereo microphone system, which was developed in [9], is also integrated. Once the head and hands of a person have been detected, the humanoid robot mimics the head and hand motion of the person. When the robot loses the person, the attention of the robot is regained through the acoustic localization. Our experiments indicate a robust visual and acoustic tracking of a person even when the head and the platform are moving.

## 6 Conclusion

The paper has introduced a hierarchical control architecture for humanoid robots as well as a framework for the coordinated execution of humanoid robots tasks. A Petri net based coordination strategy has been presented and several manipulation and locomotion experiments have been carried out to prove the suitability of the proposed strategy. The communication between different controllers of the robot and the control mechanism for coordinated task execution is efficiently specified within the Petri net framework. The papers shows that Petri nets can be used as an efficient tool for the coordinated motion control in robotic systems with a high degree of freedom such as humanoid robots. Many functionalities were integrated through an event-based coordination scheme in order to realize various tasks in household environments.

**Acknowledgment**

This work has been performed in the framework of the german humanoid robotics program SFB 588 (project R1) funded by the *Deutsche Forschungsgemeinschaft (DFG).*

## References

1. R. David and H. Alla, “Petri Nets for Modeling of Dynamic Systems,” *Automatica*, vol. 30, pp. 175–202, 1994.
2. T. Asfour and R. Dillmann, “Human-like Motion of a Humanoid Robot Arm Based on Closed-Form Solution of the Inverse Kinematics Problem,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), 27-31 October, 2003.
3. T. Asfour, K. Berns and R. Dillmann, “The Humanoid Robot ARMAR: Design and Control,” The 1st IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2000), 7-8 September, 2000.
4. R. A. Brooks, “A Robust Layered Control System for a Mobile Robot,” IEEE Journal on Robotics and Automation, vol. RA-2, pp. 14–23, 1986.
5. C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems,* Kluwer Academic Publ. 1999.
6. P. Hsu, “Control of Multi-Manipulator Systems: Trajectory Tracking, Load Distribution, Internal Force Control and Decentralized Architecture,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1234–1239, 1989.
7. S. Arimoto, F. Miyazaki and S. Kawamura, “Cooperative Motion Control of Multiple Arms or Figures,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1407–1412, 1987.
8. O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A. Casal, “Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation,” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96), vol. 2, pp. 546 – 553. 1996
9. D. Bechler, M. Schlosser and K.Kroschel. “Acoustic 3D Speaker Tracking for Humanoid Robots with a Microphone Array,” *Proceedings of the 3rd IEEE International Conference on Humanoid Robots (Humanoids 2003)*, Karlsruhe, Germany.
10. Y-J. Cho, J-M. Park, J. Park, S-R. Oh and C. W. Lee “A Control Architecture to Achieve Manipulation Task Goals for a Humanoid Robot,” *Proceedings of the IEEE Intern. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 206–212, 1998.
11. K. Nickel and R. Stiefelhagen. “Pointing gesture recognition based on 3d-tracking of face, hands and head orientation,” *International Conference on Multimodal Interfaces*, Vancouver, Canada, 2003.
12. G. N. Saridis and K. P. Valvanis, “Analytic Design of Intelligent Machines,” *Automatica*, vol. 24, pp. 123–133, 1988.
13. F. Y. Wang, K. J. Kyriakopoulos, A. Tsolkas and G. N. Saridis, “A Peti Net Coordination Model for an Intelligent Mobile Robot,” *IEEE Transactions on System, Man and Cybernetics*, vol. 21, no. 4, pp. 777–789, 1991.
14. T. J. Tarn, A.K. Bejczy and X. Yun, “Coordinated Control of Two Robot Arms,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1193–1202, 1986.
15. ATI Industrial Automation Homepage: http://www.ati-ia.com/sensors.htm
16. Modular Controller Architecture (MCA2): http://mca2.sourceforge.net/
17. Sick-Homepage: http://www.sick.de/de/products/