A Robotic Soccer Passing Task Using Petri Net Plans

(Demo Paper)

P.F. Palamara
Dipartimento di Informatica e Sistemistica
Rome, Italy
pierpal@gmail.com

V. A. Ziparo
Dipartimento di Informatica e Sistemistica
Rome, Italy
ziparo@dis.uniroma1.it

L. Iocchi
Dipartimento di Informatica e Sistemistica
Rome, Italy
iocchi@dis.uniroma1.it

D. Nardi
Dipartimento di Informatica e Sistemistica
Rome, Italy
nardi@dis.uniroma1.it

P. Lima
Institute for Systems and Robotics
Lisboa, Portugal
pal@isr.ist.utl.pt

H. Costelha
Inst. for Systems and Robotics
Lisboa, Portugal
Polytech. Inst. of Leiria, ESTG
Leiria, Portugal
hcostelha@isr.ist.utl.pt

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation*

General Terms

Design, Experimentation

Keywords

Petri Nets, Multi-robot plans, Distributed Execution

1. INTRODUCTION

This demonstration shows the implementation of a cooperative and coordinated robotic task in the soccer domain. Two Sony AIBO robots are placed on a soccer field with the task of passing a ball (Figure 1). Initially the robots search for the ball, and when they have located it, they exchange their local information about its position. This information is used to decide which robot will pass the ball (i.e. take the $Passer$ role) and which robot will intercept it (i.e. take the $Receiver$ role). The $Passer$ robot reaches the ball, grabs it and turns to the $Receiver$. In the meanwhile, the $Receiver$ moves to a predefined distance to the $Passer$. When both the $Passer$ and $Receiver$ robots have completed these tasks, they synchronize and execute, respectively, a pass and an intercept behavior. At the end of the execution, a final synchronization is performed. The robots exchange information about the outcome of the behavior, and they repeat the plan.

The complexity of this task requires not only basic robot functionalities, as object recognition, position estimates and basic behaviors, but also a formalism for high level representation and execution of multi-robot plans. To this end, we adopt multi-robot PNPs [2], that allow the representation of high level multi-robot programs, providing all the features needed to describe complex plans in dynamic, partially observable and unpredictable environments. In particular, multi-robot PNPs provide the means for deliberate cooperation, synchronization of actions among different robots and cooperative handling of local failures.

Multi-robot PNPs implement a centralized planning for distributed execution approach, by automatically dividing multi-robot plans in single robot ones. Each single robot plan is executed independently relying on local knowledge. Moreover, during the plan execution, robots can communicate through a reliable channel to attain synchronization and sharing of information.

Figure 1: A picture of the pass task executed by two Sony AIBO robots

In the following we detail the critical parts of the multi-robot PNPs necessary for accomplishing the passing task. The operators and the notation used for the presentation are described in [2]. The complete multi-robot plans, derived single robot plans and videos showing the execution of this task are available at [1].

2. COORDINATION AND COOPERATION

Two critical aspects of this passing task are the coordinated assignment of the two cooperative roles and the synchronization required to successfully perform the pass.

The assignment of the roles ($Passer$ and $Receiver$) is accomplished in the multi-robot plan through a *hard synchronization* operator, followed by two concurrent sensing actions, as shown in Figure 2. After the ball has been successfully located by the two robots, the h_sync operator is used to synchronize the execution and to exchange information about each robot's estimated distance from the ball. This communication ensures that both robots share the same set of beliefs that are required for coordination. The assignment is then consistently performed through the sensing of the condition $closestToBall$. In case $Robot1$ is the closest to the ball ($R1.closestToBall$ is $true$), the robot will grab it and perform a

Cite as: A Robotic Soccer Passing Task Using Petri Net Plans (Demo Paper), P.F. Palamara, V. A. Ziparo, L. Iocchi, D. Nardi, P. Lima, H. Costelha, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16.,2008,Estoril,Portugal,pp.1711-1712.
Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

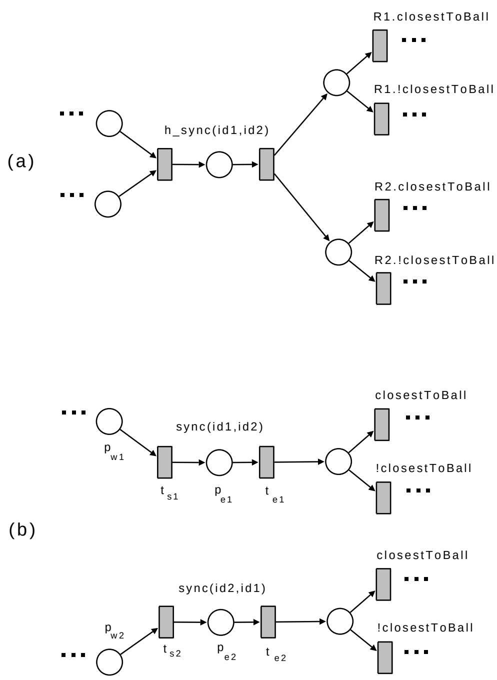

Figure 2: Petri Net Plan used for task assignment: (a) multi-robot plan (b) single-robot plans

pass towards $Robot2$. The pass and receive procedures are encoded in the remaining branches of the plan, not shown in the figure. The h_sync operator relies on the single-robot $sync$ primitives, that make use of the underlying network protocol to achieve communication. The presence of a token in the place p_{w1} causes $Robot1$ to send to $Robot2$ the unique id_1, which encodes its plan execution state. The transition t_{s1} fires at the reception of id_2, sent by $Robot2$, that acts symmetrically. A token in the execution places p_{e1} and p_{e2} denotes that the robots are performing the communication. When the information about the distance from the ball has been successfully sent, the transitions t_{e1} and t_{e2} fire.

Synchronization is not only needed in this task for assigning the two roles. The execution of the pass behavior requires the robots to synchronously perform their actions. When the $Passer$ robot terminates its rotation towards the $Receiver$, and when the $Receiver$ reaches a desired position, a synchronization is performed in order to simultaneously start the passing and the interception of the ball. A final synchronization has been inserted after these actions, allowing the robots to exchange useful information about the outcome of the performed task (e.g. success or failure of the ball interception). Figure 3 shows the use of the h_sync operator in the multi-robot plan at the end of the preparation phase.

3. FAILURE HANDLING

The execution of the multi-robot plan for the passing task is subject to failures, as unpredictable events may occur during the performance of the actions. Indeed, the implementation of effective passing primitives on AIBO robots is a complicated task, and the multi-robot plan must allow the detection of action failures at execution time. Detected local action failures must be communicated and cooperatively handled. In this passing task, during the rotation, the $Passer$ robot grabs the ball between its front legs. In case the robot detects a ball in the field during this phase, the $Receiver$ robot is notified and the execution is interrupted. Figure 4 shows

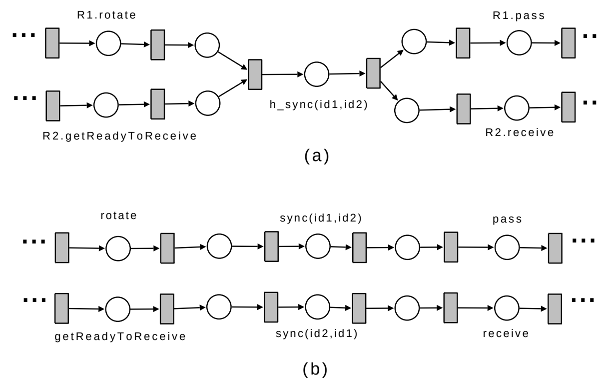

Figure 3: Synchronization operator used after the preparation phase: (a) multi-robot plan (b) single-robot plans

a portion of the multi-robot plan used for the interruption. The $lostBall$ condition is activated in case the $Passer$ detects the failure during the rotation phase. The single agent decomposition (Fig. 4-b), shows how the multi-robot interrupt is seen as a local interrupt operator by the $Receiver$ robot.

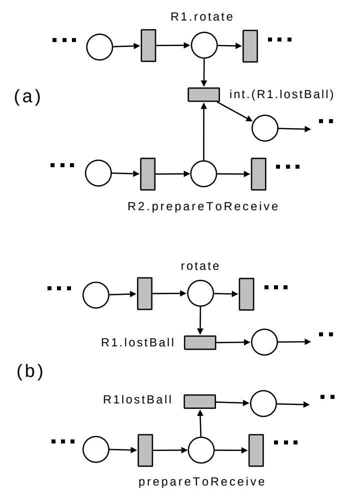

Figure 4: Petri Net Plan used for action interruption: (a) multi-robot plan (b) single-robot plans

As for the case of the $sync$ primitives, the firing of the interrupt transitions in the single-robot plans is caused by the communication of a unique interrupt id, sent by the $Passer$ robot if the $lostBall$ condition is verified. The structure shown in Figure 4 is merged in the final plan with the one shown in Figure 3, where the interrupt operator was not included for readability.

4. REFERENCES

[1] Extra Material Web Page. `www.dis.uniroma1.it/~ziparo/pnp_extras.html`.

[2] V. Ziparo, L. Iocchi, D. Nardi, P. Palamara, and H. Costelha. Pnp: A formal model for representation and execution of multi-robot plans. In M. Padgham, Parkes and Parsons, editors, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Estoril, Portugal, May 2008. IFAAMAS Press.