Formalizing Institutions as Executable Petri Nets for Distributed Robotic Systems

José N. Pereira[1,2], Porfírio Silva[2], Pedro U. Lima[2] and Alcherio Martinoli[1]

[1] Distributed Intelligent Systems and Algorithms Laboratory, EPFL, Lausanne, Switzerland
[2] Intelligent Robots and Systems Group, Institute for Systems and Robotics, Lisbon, Portugal
[1] {jose.pereira | alcherio.martinoli}@epfl.ch, [2] {porfiriosilva | pal}@isr.ist.utl.pt

Abstract

Institutional Robotics is a new approach to the coordination of distributed robotic systems, drawing inspiration from social sciences. It aims to provide a comprehensive strategy for specifying social interactions among robots in the form of institutions. In this paper, we present a formalism for institutions in the Institutional Robotics model. We apply this formalism to two case studies. The first is concerned with a swarm of simple robots which has to maintain wireless connectivity. The second focuses on role allocation in a robotic team aimed at improving coordination and performance in a transportation task.

Introduction

Multi-robot systems are nowadays an important area of research within the broader field of robotics. Using multiple robots might enhance the overall system performance not only because of a faster task execution speed but also in terms of robustness to failures and flexibility in allocation of subtasks. It is also clear that a team of robots is capable of completing some tasks that are impossible for single robots, for instance, because of their physical limitations. However, in order to leverage these potential benefits, it is not enough to add robots to the team. Cooperative behavior has to be present, and therefore interactions among robots must be coordinated in some way.

Institutional Robotics (IR) (Silva and Lima (2007)) is a new approach to the coordination of distributed robotic systems, drawing some inspiration from social sciences, namely from Institutional Economics' concepts. It combines the notions of institution, coordination artifact, and environment, aiming to provide a comprehensive strategy for specifying social interactions (e.g., norms, roles, hierarchies) among robots. In order to do so, robots are situated not only in a physical but also in an institutional environment, where their interactions are guided by institutions. Through cooperative decision-making, these institutions can be modified by the robots, providing adaptation to a changing scenario. Coordination is achieved by this regulation of social interactions since the robots know not only how to behave in a given scenario but also what to expect from other robots and the environment.

One of the goals of our research is to formalize the concepts of IR from a computer science perspective, so as to create an ontology of the entities that will be part of the IR model, and to describe ways of interconnecting them (such as graphs and tuples describing the entities associated to each node), as well as algorithms to manage a robotic collective based on social science principles.

In this work, we focus on formalizing the central concept of IR - institutions. Institutions are coordination artifacts specifying social interactions of different types and encapsulating relevant behavioral rules (possibly designed based on problem-domain knowledge) that, once adopted, avoid the need for the behavior to be re-learned or re-acquired. Our goal is to formalize them using an abstract representation, that will allow us to design these coordination artifacts and execute them in robots (both in reality and simulation), so as to obtain behaviors capturing the social interactions of interest. In order to accomplish this objective we propose to use Petri Nets as an abstract representation for institutions. Our method will produce, from a set of institutions, a robot controller able to execute a desired task.

We apply this formalism to two case studies. The first is concerned with a swarm of simple robots which has to maintain wireless connectivity. The second focuses on role allocation in a robotic team aimed at improving coordination and performance in a transportation task.

In Section 2 we discuss related work and motivation for our formalization. This formalization is presented in Section 3 culminating with the definition of a controller based on our institutional approach. In Section 4 and 5 we apply this formalism to two different case studies.

Related Work

Institutional economics is a fundamentally different approach from neo-classical theory, the current trend of economics and inspiration for market-based systems of task allocation in distributed robotics (Dias et al. (2006)).

In Hodgson (2000), the author refines a description of institutional economics outlining the following main features: institutions are the key element of any economy; the economy is an open and evolving system; and the notion of individuals as utility-maximizing agents is inadequate. The institutional approach is characterized also by the rejection of unbounded rationality. Agents are affected by the institutional environment they live in, but in no way does that environment fully determine their behavior. Every agent has individual goals and motivations that it wants to fulfill. Institutions are developed by these very same agents.

In Crawford and Ostrom (1995) and Ostrom (2005), the authors propose a formal "grammar" of institutions according to the New Institutional Economics (NIE) approach. NIE is a compromise between the institutional and neoclassical theories of economics. Therein, the authors study what are the elements that compose institutional statements. While at this point most of these elements are not ready to be applied to multi-robot systems, deontic operators are fundamental in our IR version in order to specify how institutions relate to one another.

IR (Silva and Lima (2007)) aims to provide a comprehensive strategy for specifying social interactions among robots, by combining the notions of institution, coordination artifact, and environment. According to the IR approach:

1. the coordination strategy is supported by a network of institutions;
2. institutions are coordination artifacts of different types (e.g., norms, roles, hierarchies);
3. robots are able to modify both their physical and their institutional environment;
4. robots need a high degree of autonomy, pursuing goals based on their "struggle for survival".

From an institutional perspective, institutions are taken as the main tool of any sophisticated society, and individuals are both constructive within and constructed through institutional environments. In a first attempt at formalizing institutions in the IR model, Silva et al. (2008) define them as "cumulative sets of persistent artificial modifications made to the environment or to the internal mechanisms of a subset of agents, thought to be functional to the collective order".

This definition is too abstract to be applied "as is" to distributed robotics experiments. Thus, we go back to the idea of institutions as coordination artifacts (Tummolini and Castelfranchi (2006)). Coordination artifacts (Omicini et al. (2004); Ricci et al. (2005)) are infrastructure abstractions in multi-agent systems meant to improve the synthesis and analysis of coordination activities. The main properties that describe coordination artifacts are: *specialization*, *encapsulation*, and *inspectability*. Specialization refers to the fact that coordination artifacts are specialized in automating coordination activities and can be represented with concurrency frameworks such as Petri Nets or process algebras. Coordination artifacts encapsulate a coordination service, allowing the agents to abstract how it is implemented. Encapsulation is the key to achieve reuse of coordination. Inspectability refers to the property that an artifact should support some procedure to allow engineers or agents responsible for the system to check for errors in its specification.

Omicini et al. argue that coordination artifacts are exterior to the agents using them and perceived as individual entities, but can actually be distributed on several nodes of a multi-agent system. We propose that, when taking institutions as coordination artifacts, they can be part of the agent controller, working as norms or procedures the agent has to follow. Even with this assumption, we can still think of institutions being distributed in our multi-robot system, if we consider their representation to be replicated in each agent.

Petri Nets and Institutions

Starting from the concept of institutions as coordination artifacts we model them using a formal representation, leading to a standard design and execution platform (in real robots, realistic simulations, and multi-agent systems). Considering the three main properties of coordination artifacts mentioned above, we propose to use Petri Nets as formal framework.

Our choice of Petri Nets is based mostly on the ability of this formalism to deal with distributed systems. State information is distributed among a set of places that capture key conditions that govern the operation of the system. Moreover, Petri Nets not only are able to deal with distributed systems but are also a suitable computational model for effective and efficient interaction management, a key aspect of coordination artifacts. Finally, Petri Nets also have a larger representational power than Finite State Automata (FSA), being able to represent, with finite structure, languages that are not representable by FSA (Cassandras and Lafortune (2008)).

The Petri Net Plans (PNP) language is a tool specifically directed to the design and execution of robotic plans using Petri Nets (Ziparo et al. (2010)). Therein, properties of safety and liveness of PNs are used to ensure that execution of robotic tasks in robots follows the designed plan. However, these properties can also be verified on simpler Petri Nets models without the need of using the PNP methodology, which can be restrictive on the types of tasks that can be designed.

A multi-layer methodology, introduced in Costelha and Lima (2010), enables organizing separately the interaction between multiple institutions and the behavior of the robot as a single individual (which we will hereafter call "individual behavior"). While this is achieved in a higher layer, the execution of each institution can be described in a lower layer and represented on the above layer by means of macro places. By using Costelha and Lima (2010) expansion algorithm we can obtain a full Petri Net that can be tested for our desired properties. Also, this will allow us to add more institutions on-the-fly (during the robots execution) and still maintain these properties.

Executable Petri Nets

We follow the definitions for Petri Nets and their dynamics (enabled transitions, state transition dynamics) in Cassandras and Lafortune (2008):

Definition: A *Petri Net* is a five-tuple (P, T, A, w, X) where:

- P is the finite set of *places*;
- T is the finite set of *transitions*;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs from places to transitions and from transitions to places;
- $w : A \rightarrow \mathbb{N}^+$ is the *weight function* on the arcs;
- X is a *marking* of the set of places P, $X = [x(p_1), \ldots, x(p_n)] \in \mathbb{N}^n$ represents the *state* of the Petri Net.

Herein, we assume that all the weights of the arcs are 1. If $x(p_i)$ in marking X is equal or larger than 1, we say that place p_i is marked. Each unit in $x(p_i)$ is called a *token*, i.e., if $x(p_i) = 1$ then p_i has one token. State transitions in Petri Nets occur by moving tokens through the net and changing the marking by doing so. The sets of input places $I(t_j)$ and output places $O(t_j)$ of a transition t_j are given by $I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}$ and $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$. Petri Net dynamics are provided by the following state transition function:

Definition: The *state transition function*, $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$, of Petri Net (P, T, A, w, X) is defined for transition t_j if and only if

$$x(p_i) \geq w(p_i, t_j) \;\; \text{for all } p_i \in I(t_j) \tag{1}$$

If $f(X, t_j)$ is defined, then we set $X' = f(X, t_j)$, where

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i), \;\; i = 1, \ldots, n \tag{2}$$

If transition t_j verifies condition (1) then we say it is *enabled*. When transition t_j is enabled, we say that it can *fire*, and thus trigger a state change on the net by moving tokens according to (2).

Our aim is to formalize institutions as Petri Nets both for design and execution of robotic controllers. This means that we need to take into account robot actions and sensor readings. We consider three sets of building blocks that will allow us to design our controllers.

The set Act contains all robot primitive actions (combinations of two or more primitive actions are not considered as primitive actions).

The set Cdt contains boolean conditions that can be verified by checking sensor readings.

Finally, the set Pac contains “parameter actions”, which are auxiliary actions not concerning actuators but that only modify variables needed for the actions in Act.

We are now able to define our own version of Petri Nets used for execution of our robotic controllers.

Definition: An *Executable Petri Net* (EPN) is a Petri Net (P, T, A, w, X) where:

- each place $p_i \in P$ has an associated action $a_i \in Act$;
- each transition $t_i \in T$ has an associated condition $c_i \in Cdt$ and an associated parameter action $pa_i \in Pac$.

The basic intuition behind this definition is that by associating actions with places we are able to define which actions are to be executed at each time step. This is done simply by checking if the corresponding place is marked. By associating transitions with conditions verified by sensor readings we trigger state changes in the Petri Net due to changes in the robots environment. The following algorithm is performed by the robots at each time step, allowing the robots to execute the behavior designed in an EPN.

Algorithm 1 Execute Petri Net

```
repeat
    for all enabled transitions $t_i \in T$ do
        if associated condition $c_i$ is true then
            run associated parameter action $pa_i$
            fire transition $t_i$
        end if
    end for
until no transition has fired
for all marked places $p_i \in P$ do
    run associated action $a_i$
end for
```

The implementation code for actions and conditions present in the sets Act, Cdt and Pac is not explicitly represented in the code that specifies an EPN. All robots share a common function table that implements all possible actions and conditions. These are then represented in the EPN by means of indices. This allows the EPNs to be generic, in a sense that although robots may have different implementations for the same action (e.g., heterogenous robots in terms of hardware), the same EPN could be used to achieve coordination in the same manner. Also, it enables the sharing of EPNs among robots without the sharing of the actual implementation of actions.

Institutional Agent Controller

Our goal is to formalize institutions as coordination artifacts in a modular fashion. We intend to have each institution represented by an EPN that can be executed independently or together with other institutions. The *individual behavior* for the robots is also represented by an EPN. While the institutions specify behaviors that have a *social nature*, i.e., they relate the robot to other robots in some way, the individual behavior specifies a set of basic behaviors that have exclusively an *individual nature*, i.e., they relate the robot with the surrounding environment. The composition of the individual behavior with a set of institutions will generate a robot controller.

We now present our formalized definition of institution:

Definition: An *Institution* I is a four-tuple $(Inst, initial_I, final_I, d_I)$ where:

- $Inst$ is an EPN;
- $initial_I, final_I \in Cdt$ are initial and final conditions for the execution of $Inst$;
- $d_I \in D$ is the associated deontic operator.

The EPN $Inst$ specifies the desired behavior that should be performed by the robot. This behavior is not always being executed, its start and finish are dictated by conditions $initial_I$ and $final_I$, which the robot verifies at each time step. Thus, we say that an institution I at each time step can be *active* or *idle*. Each institution also includes a deontic operator d_I which is used when combining it with the robot individual behavior and further institutions. Despite $Inst$ being designed by hand, institutions can be kept simple (e.g., arc weights set to 1) and further behavioral complexity can reached by composition, in a modular fashion.

A previous abstract definition of institution was presented in Silva et al. (2008). There, the authors define the institution as a tuple *(ID, Rationale, Modifiers, Network, Institutional Building, History)*, where each element of the tuple tries to capture the main constitutive elements of the social order dynamics. For our purpose of formalizing institutions using an abstract representation, allowing for a standard design and execution platform, this definition is not sufficient. However, the EPN $Inst$ can be seen as part of $Rationale$, since it specifies the activity of the institution, and the deontic operator as part of $Network$, since it specifies how the institution relates to other institutions.

The composition of the individual behavior with a set of institutions is non-trivial since concurrent execution of some of the institutions might be impossible or at least inadequate to the task the robot is carrying out. An example of such institutional interplay is that an institution stating that you must drive on the right side of the road will be overruled by the institution of the road code of Great Britain, and thus should not be executed when in that territory. Crawford and Ostrom (1995) define a set of deontic operators, $D = \{P, O, F\}$, establishing permitted (P), obliged (O), and forbidden (F) operations, to be applied to institutional statements in order to deal with this problem. In our formalization, these operators affect whether institutions are active or idle at each time step. However, the conditions that govern when a specific institution is active might refer directly to the activity state of other institutions. For instance, the institution for driving on the right is forbidden (and thus should be idle) when the institution of the road code of Great Britain is active. This referencing of other institutions creates a problem for our intended modular approach to formalization. Therefore, we have chosen to use a more restrictive set of deontic operators in order to guarantee that institutions do not refer to any other specific institution but can still prevent the concurrent execution of undesired behaviors (individual behavior and other institutions in general).

Definition: The set D of deontic operators for IR institutions includes the following deontic operators: $\{AllowAll, StopInd, StopInst, StopAll\}$. Their corresponding definitions are as follows:

- $AllowAll$ implies that the associated institution can be executed concurrently with the individual behavior and all the other institutions;
- $StopInd$ implies that the associated institution cannot be executed concurrently with the individual behavior;
- $StopInst$ implies that the associated institution cannot be executed concurrently with other institutions;
- $StopAll$ implies that the associated institution cannot be executed concurrently with the individual behavior or other institutions.

Herein we define the individual behavior simply as an EPN Ind.

As previously mentioned, Petri Nets (and thus EPN) can be represented by macro places in a hierarchical fashion, using two distinct layers. We consider that individual behavior and institutions are part of a lower layer and are represented by one macro place in the higher layer, as shown in Fig. 1. On the left side (lower layer) the EPN $Inst$ of institution I is displayed. On the right side (higher layer) the macro place m_I representing institution I is displayed. By adding arcs from each transition in $Inst$ to m_I and from m_I to each transition (shown as a single bidirectional dotted arc), we guarantee that each transition will only be enabled if m_I is marked. When a transition in $Inst$ fires, m_I will continue to be marked since it is a output place of the transition.

Thus, if a macro place is marked, the individual behavior or institution that it represents is active, otherwise it is idle. This allows us to compose our institutions in the higher layer where relationships among the institutions and the individual behavior should be specified, while keeping relationships between actions and conditions separated in the lower layer. Both layers can be then merged algorithmically

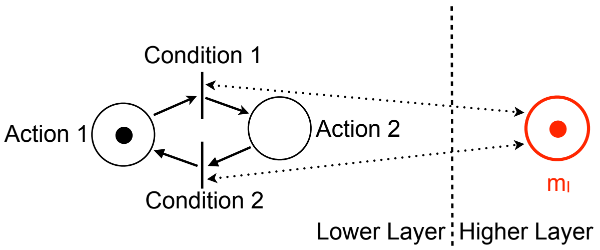

Figure 1: Hierarchical representation of an EPN in two layers. Dotted arcs represent two directional arcs, one from a transition to a place and one from a place to a transition. Left side: lower layer, EPN $Inst$ with conditions and actions associated to transitions and places. Right side: higher layer, macro place m_I in red.

(Costelha and Lima (2010)) to obtain a full EPN that can be used as controller.

To understand how the composition of institutions is made, we consider a minimal setup with two institutions I_1 and I_2 and an individual behavior Ind. A representation of the higher layer of this setup before composition is presented in Fig. 2-(a). Places in red (m_{I1}, m_{I2}, m_{Ind}) represent in the higher layer institutions (I_1, I_2) and the individual behavior (Ind) implemented at the lower layer. Places $idle_{I1}$ and $idle_{I2}$ further represent the idea that institution I_i is active if place m_i is marked. Since only one place from the set m_i and $idle_i$ can be marked at each time, we have that institution I_i is active if m_i is marked and idle if $idle_i$ is marked. This allows us to regulate the activation and idling of institutions with their initial and final conditions as shown in the Fig. 2-(a). The individual behavior does not have an idle place since it has no initial or final conditions.

The composition of individual behavior and institutions is controlled by the deontic operators as presented in Fig. 2. As stated before, composition takes places only in the higher layer. We will see how different deontic operators for institution I_1 control the composition while always maintaining the deontic operator of institution I_2 as $AllowAll$. If the deontic operator of institution I_1 is also $AllowAll$ (Fig. 2-(a)), then no other relationship is necessary since all behaviors can be executed concurrently. If the deontic operator of I_1 is $StopInd$, the structure in Fig. 2-(b) is added. Place $idle_{Ind,I1}$ represents the individual behavior being idle because of institution I_1 being active. The added transitions have associated a special condition that is always true. This specifies that if institution I_1 is activated, then the individual behavior is set to idle and vice-versa. If the deontic operator of I_1 is $StopInst$, as in Fig. 2-(c), the same structure is added but now related to the macro places of the other institution and not the individual behavior. Our setup considers only two institutions but the structure would be added for all institutions except I_1, if more institutions were present. This means that institution I_2 can be idle if place $idle_{I2}$ is marked or if place $idle_{I2,I1}$ is marked. On the latter case, institution I_2 will resume being active when institution I_1 becomes idle. If the deontic operator is $StopAll$ then we consider a combination of the previous two cases, as show in Fig. 2-(d). These rules also apply for institution I_2 if it has a different deontic operator than $AllowAll$.

We can now define our Institutional Agent Controller that will guide the performance of our robots:

Definition: An *Institutional Agent Controller* (IAC) is an EPN resulting from the composition of an individual behavior Ind and a set of institutions $\{I_1, \ldots, I_n\}$ controlled by the deontic operators $d_{I_1}, \ldots, d_{I_n}$.

All macro places and control places ($idle_i$) added during composition are associated with a void action. Considering these associations, our IAC is itself an EPN and can be executed by Algorithm 1. A minor change is needed to line 9 of the algorithm to make sure that not only the lower layer place is marked but also the higher layer macro place of the institution being executed. Time needed for the formalization includes the design time of the institutions and individual behavior and composition time. While the latter is performed algorithmically with negligible time, the former requires a certain amount of time and experience with design of behavior-based controllers (the same as with FSA).

The IAC for a desired task can be obtained prior to an experiment and transmitted to the robots. It is also possible for each robot to obtain the IAC from a given set of institutions at the start of the experiment. Thus, the method is fully scalable to any number of robots. Complexity of the IAC increases only with the number of institutions.

Wireless Connected Swarm Case study

In this section we present a case study to illustrate how to apply our formalism of institutions in order to obtain an IAC that performs the desired task. Our aim is to be able to specify behaviors that have a social nature as institutions and behaviors that have an individual nature as individual behavior.

We have selected a case study previously investigated by Nembrini et al. (2002) and Winfield et al. (2008), where a decentralized control algorithm is able to maintain a certain degree of spatial compactness of a robotic swarm (with N robots) using exclusively, as information at the robot level, the current number of wireless connections to the neighbors. The communication is local and its bounded range a parameter of the robotic system. Let X be the number of connections perceived by a robot. In the default state, the robot simply moves forward. If at any time X falls below a threshold α (where $\alpha \in \{0, \ldots, N-1\}$), the robot assumes it is going in the wrong direction and turns back. Upon X returning to a value above α, the robot performs a random turn and

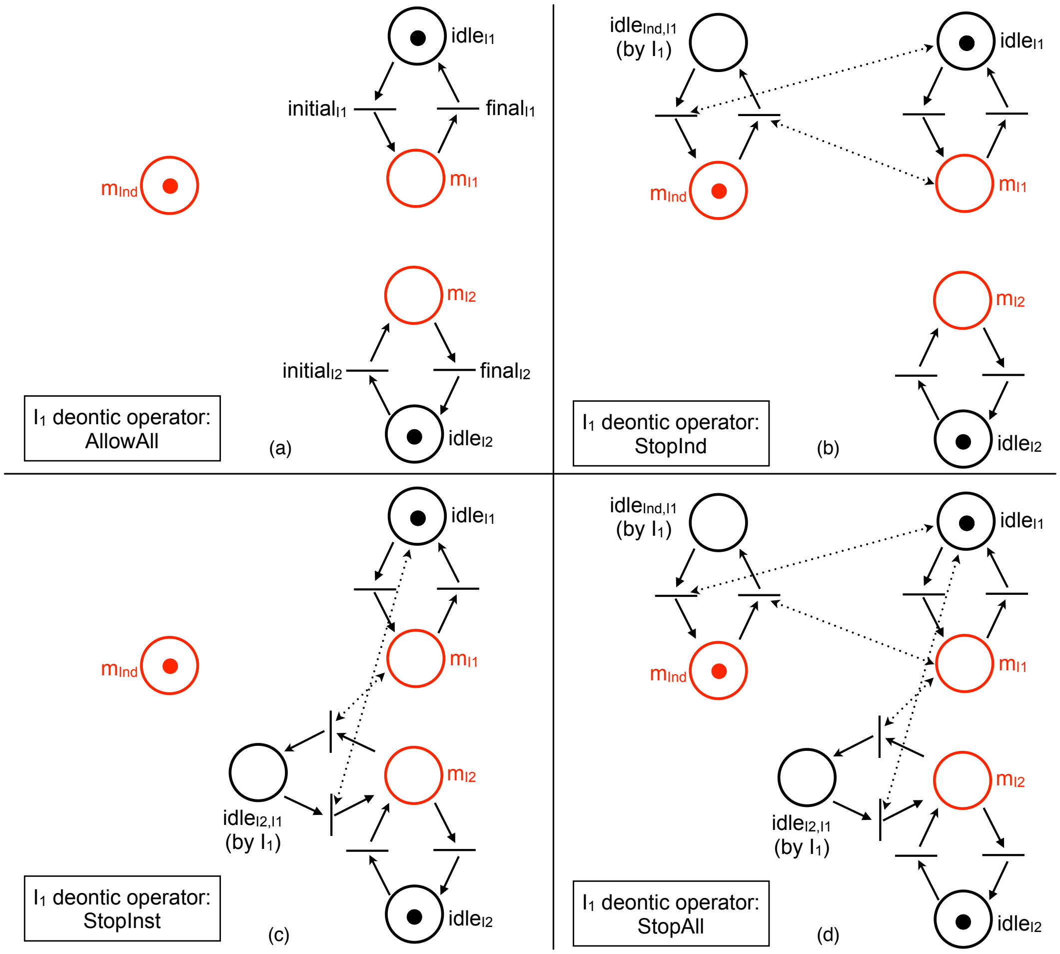

Figure 2: Composition scheme for two institutions I_1, I_2 and individual behavior Ind. Dotted arcs represent bidirectional arcs, as in Fig. 1. Places in red are macro places representing implementations of institutions and the individual behavior in the lower layer. These representations will be used throughout the paper. (a) composition rule with deontic operator $AllowAll$; (b) composition rule with deontic operator $StopInd$; (c) composition rule with deontic operator $StopInst$; (d) composition rule with deontic operator $StopAll$.

moves back to the default state. Robots always execute obstacle avoidance at the same time. This simple algorithm is quite fragile but allows the swarm to maintain its connectivity to a certain extent, with its spatial compactness being controlled by the communication range.

Our case study is similar to that of Nembrini et al. (2002) with the following differences: (i) no random turn is executed when the robots are connected again; (ii) our arena is bounded by a wall. Robots execute an individual behavior Ind and an institution I, both specified by EPNs with only two places shown in the left side (lower layer) of Fig. 3. Individual behavior Ind consists of a simple obstacle avoidance. Robots move forward until they find an obstacle (wall or other robot), perform a turn with random degree and return to moving forward. Institution I implements the social rule, specifying that when a robot loses connections below α it should turn back.

To consider the institution as defined in Section 3, we need initial and final conditions and a deontic operator. We say that initial condition $initial_I$ is “number of connections is less than α” and the final condition $final_I$ is “turn 180° procedure has ended”. The associated deontic operator is $StopInd$ specifying that institution and individual behavior cannot be executed concurrently.

We now have all the elements needed to obtain the IAC that specifies our desired behavior. The composition of the individual behavior Ind and institution I on the left side (lower layer) of Fig. 3 is shown in the right side (higher layer) of Fig. 3. The final controller is the full EPN of Fig. 3 after the merging of the two layers. Lower layer actions and

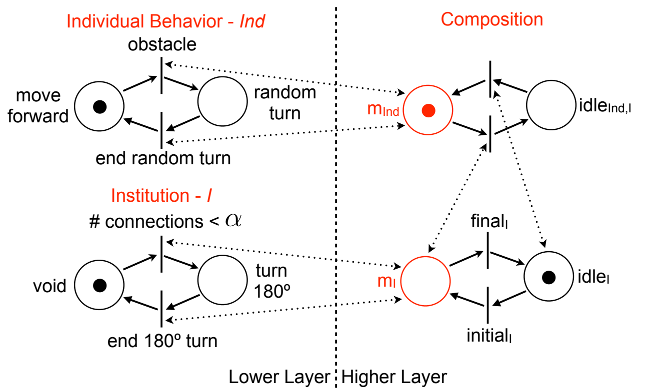

Figure 3: IAC for wireless connected swarm. Left side: lower layer EPNs for individual behavior Ind and institution I. Right side: EPN resulting from composition of individual behavior Ind and institution I.

conditions are implemented in the robot. Thus, to perform the task the robot needs only to execute Algorithm 1 taking the IAC as input. Actions associated with marked places are executed, much in the same manner as in a FSA actions associated with states would be executed.

Corridor Case Study

A previous study concerning the institutional approach was presented in Pereira et al. (2010). Therein, institutional robotics concepts were taken into account when developing a controller for robots that had to coordinate their movement in order to traverse a narrow corridor while performing a simple transportation task. However, no formalization of the IR approach was proposed in that study. Again, our aim is to specify behaviors that have a social nature as institutions and summarize behaviors that have an individual nature as the robots' individual behavior. Our setup will consider two institutions and the individual behavior. As this case study is of higher complexity than the previous one, due to space limitations, we will not be able to describe the EPN implementations in its completeness. Therefore, we will focus only on the higher layer of the IAC.

The task consists of transporting a virtual payload in an arena with two rooms connected by a corridor. Navigation of the robots is done by performing a wall-following behavior. Transporting robots pick up the virtual payload in the left room. They must then navigate through the corridor and deploy the payload in the right room. This is the individual behavior Ind of the robots.

The corridor connecting the rooms is too narrow for two robots moving in opposite directions to pass one another. Thus, the robots must traverse the corridor in one direction at a time. Robots need to cooperate to avoid collisions and deadlocks in the corridor. In order to facilitate coordination, we let a subset of the robots adopt the institutional role of "traffic regulators" to control the circulation of the remaining robots in the team. The overall traffic regulation implies robots serving as regulators and robots accepting to give priority to others in case the regulators will ask them to do so. We will therefore need two institutions, one to manage the allocation and execution of the role of regulator, and one to receive information about priority from the regulators.

If the need of traffic regulating robots arises due to a physical conflict between two robots in the corridor, these very same robots assume the role as traffic regulators. The two traffic regulators place themselves at the opposite ends of the corridor so that each regulator can control the flow of transporting robots entering the corridor from one of the rooms. The goal of the regulators is to ensure that robots only move through the corridor in one direction at a time. The regulating robots are synchronized so that only one of them will let transporting robots enter the corridor from their respective rooms at any given time. The regulation is performed by sending stop and go messages to the transporting robots.

This is clearly a behavior that has a social nature. We consider that this behavior corresponds to an institution I_R that manages the role of traffic regulator. Its initial condition $initial_R$ is the detection of a conflict in the corridor and its final condition $final_R$ is the end of regulation (time limit). Since we do not want this behavior to be executed concurrently with any other behavior, the deontic operator of institution I_R will be $StopAll$.

If a transporting robot receives a message to stop, it will stop in order to give priority to the robots traversing the corridor from the opposite direction. It will also begin to relay the stop message so other transporting robots behind it will stop too. As a result, the transporting robots will form a queue. When a robot in the queue receives a message to proceed, it forwards the message to any robots that may be behind it. After receiving and relaying the message the robot has priority and will traverse the corridor.

This is again a behavior that has a social nature. The behavior corresponds to an institution I_M that manages the reception and relay of messages. Its initial condition $initial_M$ is the reception of a stop message and its final condition $final_M$ is the reception of a go message. We do not want this behavior to be executed concurrently with the individual behavior, so its deontic operator will be $StopInd$.

In Fig. 4 we show the result of the composition of our two institutions and individual behavior. The IAC for this case study will be the result of merging this EPN with those on the lower layer.

Conclusion and Future Work

In this work we introduced an extension to the Petri Net formalism, Executable Petri Nets. These EPN have associated actions and conditions that allow them to be executed in robots through an algorithm presented in the paper. We de-

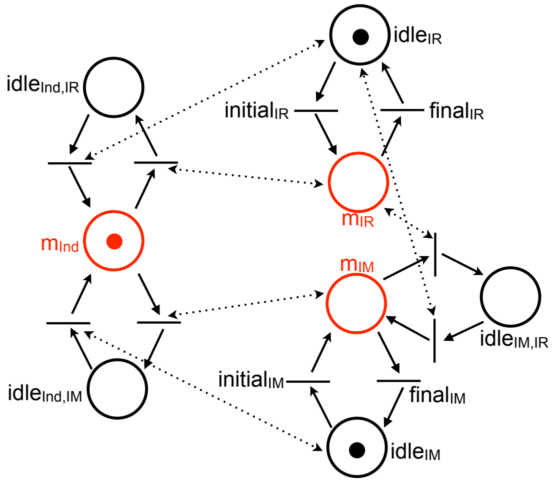

Figure 4: Higher layer EPN for corridor study. Place m_{Ind} represents the individual behavior Ind. Place m_{IR} represents institution I_R. Place m_{IM} represents institution I_M.

fined institutions and an individual behavior for robots in a distributed robotic system making use of this new extension. In our approach, institutions are modular behaviors that can be specified through an EPN and executed in a robot. Using a composition scheme controlled by dedicated deontic operators of a set of institutions we are able to obtain an Institutional Agent Controller (IAC) in the form on an EPN that combines several institutions and an individual behavior.

We applied this formalism to a simple case study where robots have to maintain wireless connections with their neighbors. We also applied the formalism to a more complex case study dealing with institutional concepts, in this case, the institutional role.

In the future we wish to study how our formalism of institutions with EPN allows us to study logical properties of the controller, such as safeness and liveness. We are also interested in studying stochastic properties of the controller, such as the steady state distribution of a given EPN or throughput of transitions. To enable this study we need to further refine our formalism of institutions to allow for stochastically timed transitions. We will also study the possibility of using the IAC as a starting point for the application of a multi-level modeling methodology. Learning of institutions and corresponding EPN will also be studied.

Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through grants SFRH/BD/33671/2009 (first author) as part of the Joint Doctoral Program IST-EPFL and SFRH/BPD/35862/2007 (second author), as well as by FCT ISR/IST Pluriannual funding through the PIDDAC program funds.

References

Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, second edition.

Costelha, H. and Lima, P. U. (2010). *Petri Net Robotic Task Plan Representation : Modelling , Analysis and Execution*, pages 65–90. Austria, intech edition.

Crawford, S. E. S. and Ostrom, E. (1995). A Grammar of Institutions. *The American Political Science Review*, 89(3):582–600.

Dias, M., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-Based Multirobot Coordination: A Survey and Analysis. *Proceedings of the IEEE (Special Issue on Multirobot Coordination)*, 94(7):1257–1270.

Hodgson, G. M. (2000). What Is the Essence of Institutional Economics ? *Journal of Economic Issues*, 34(2):317–329.

Nembrini, J., Winfield, A. F. T., and Melhuish, C. (2002). Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots. *Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats*, pages 273–282.

Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., and Tummolini, L. (2004). Coordination artifacts: Environment-based coordination for intelligent agents. In *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 286–293, New York, NY, USA.

Ostrom, E. (2005). *Understanding Institutional Diversity*. Princeton University Press.

Pereira, J. N., Christensen, A. L., Silva, P., and Lima, P. U. (2010). Coordination Through Institutional Roles in Robot Collectives. In van Der Hoek, Kaminka, Lespérance, Luck, S. e., editor, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 1507–1508, Toronto, Canada.

Ricci, A., Viroli, M., Mater, A., and Università, S. (2005). Coordination Artifacts : A Unifying Abstraction for Engineering Environment-Mediated Coordination in MAS. *Informatica*, 29:433–443.

Silva, P. and Lima, P. U. (2007). Institutional Robotics. In *Proc. of ECAL 2007 - 9th European Conference on Artificial Life*, pages 157–164, Lisboa, Portugal.

Silva, P., Ventura, R., and Lima, P. U. (2008). Institutional environments. In *Proc. of Workshop AT2AI: From agent theory to agent implementation, AAMAS 2008 - 7th International Conference on Autonomous Agents and Multiagent Systems*, volume 35, pages 595–604, Estoril, Portugal.

Tummolini, L. and Castelfranchi, C. (2006). The cognitive and behavioral mediation of institutions: Towards an account of institutional actions. *Cognitive Systems Research*, 7(2-3):307–323.

Winfield, A. F. T., Liu, W., Nembrini, J., and Martinoli, A. (2008). Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2(2-4):241–266.

Ziparo, V. A., Iocchi, L., Lima, P. U., Nardi, D., and Palamara, P. F. (2010). Petri Net Plans: A Framework for Collaboration and Coordination in Multi-Robot Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–40.