UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Advancing Social Interactions Among Robots: An Institutional Economics-based Approach to Distributed Robotic Systems

José Nuno Ferreira Maia Pereira

Supervisors: Doctor Pedro Manuel Urbano de Almeida Lima
Doctor Alcherio Martinoli

Thesis specifically prepared to obtain the PhD Degree in Electrical and Computer Engineering

Draft

September 2013

Abstract

This dissertation presents several advances to Institutional Robotics (IR), an approach to the coordination of multi-robot systems that takes inspiration from the social sciences, in particular from institutional economics. This approach aims to provide a comprehensive strategy for specifying complex social interactions among a team of robots and possibly between a team of robots and human actors.

Our work advances IR on three fronts. First, we develop a methodology for distributed robotic systems based on the formalization of the concept of institution. To accomplish such formalization we introduce Executable Petri Nets, an extension to the Petri net formalism that allows the specification and execution of institutions in robots. We present a composition algorithm that allows for sets of institutions to be combined into an Institutional Agent Controller (IAC). This controller represents the institutional environment of each robot, where complex social interactions are specified. Its replication in all nodes of a distributed robotic system provides the necessary coordination.

Second, we implement and validate concepts from IR, both in simulated and real robots, for laboratory scenarios designed to put forward relevant questions about the institutional approach, comparing the results against other existing approaches for multi-robot coordination. We validate the IR methodology by replicating results obtained with another control approach in a previously introduced case study concerned with a swarm of simple robots which have to maintain wireless connectivity. A real-world implementation of this case study, with a swarm of (as many as) 40 real, resource-constrained robots, was obtained and was able to accomplish the proposed task.

We investigate two further case studies dealing with more complex social interactions. In the corridor case study, we show that institutional roles can effectively help a distributed robotic system coordinate and improve performance in a given task. In the piece assembly case study, we show that the introduction of institutions in distributed robotic tasks that involve a social dilemma (individual vs group benefits) can improve performance, efficiency and sustainability (defined as the ability of a robotic team to keep its members operational). In both case studies we compare the institutional approach with a self-organized approach and discuss which situations are more suitable to each.

Third, we introduce an approach to the probabilistic modeling of distributed robotic systems controlled by IACs. Our approach follows a multi-level modeling methodology, focusing mainly on the macroscopic level and using the IAC as a starting point for a Generalized Stochastic Petri Net (GSPN) probabilistic model. We further extend the GSPN models with the introduction of an environmental information layer, where information not present in the IAC can be represented. We apply our modeling approach to two of the proposed case studies, using both data gathered from submicroscopic simulations and estimates obtained from the physical properties of the environment as input parameters for the GSPN models. We show that this methodology is able to obtain good agreement between results at different modeling levels – macroscopic, microscopic and submicroscopic – and real robot experiments.

Keywords: Institutional Robotics, Distributed Robotics, Robotic Control, Multi-Robot Coordination, Multi-level Modeling, Executable Petri Nets, Generalized Stochastic Petri Nets, Institutional Economics, Realistic Robotic Simulation, Real Robot Experimentation

Resumo

Esta dissertação adiciona vários contributos à Robótica Institucionalista (RI), uma abordagem à coordenação de sistemas de múltiplos robôs que se inspira nas ciências sociais, em particular na economia institucionalista. Esta abordagem tem como objectivo dotar uma equipa de robôs com uma estratégia abrangente para especificar interacções sociais complexas entre os membros da equipa e possivelmente entre os membros da equipa e actores humanos.

A dissertação avança a RI em três frentes. Primeiro, foi desenvolvida uma metodologia para sistemas distribuídos de robôs baseada na formalização do conceito de instituição. Para tal, introduziu-se Redes de Petri Executáveis, uma extensão do formalismo de Redes de Petri, que permite a especificação e execução de instituições em robôs. Foi apresentado um algoritmo de composição que permite que conjuntos de instituições sejam combinados num Controlador de Agente Institucional (CAI). Este controlador representa o ambiente institucional de cada robô, onde interacções sociais complexas são especificadas. A sua presença em todos os elementos de um sistema distribuído de robôs providencia a coordenação necessária.

Segundo, foram implementados e validados conceitos da RI, tanto em simulação como em robôs reais, em cenários de laboratório desenhados para realçar questões relevantes sobre a abordagem institucionalista, comparando os resultados com outras abordagens à coordenação de sistemas de múltiplos robôs. Validou-se a metodologia da RI, replicando os resultados obtidos com outra abordagem de controlo num estudo de caso previamente introduzido em que um conjunto de robôs simples tem de manter um conjunto de ligações *wireless*. Obteve-se uma implementação deste estudo de caso num cenário real, com um conjunto de (até) 40 robôs reais de capacidades limitadas, que executa a tarefa proposta com sucesso.

Investigaram-se ainda dois outros estudos de caso que incluem interacções sociais mais complexas. No estudo de caso do corredor, mostrou-se que papéis institucionais podem ajudar um sistema distribuído de robôs a coordenar-se e melhorar a execução de uma dada tarefa. No estudo de caso da montagem de peças, mostrou-se que a introdução de instituições em tarefas robóticas distribuídas que envolvem um dilema social (benefício individual vs de grupo) pode melhorar a execução, eficiência e sustentabilidade (definida como a capacidade de uma equipa robótica para manter os seus membros operacionais). Em ambos os estudos de caso, comparou-se a abordagem institucional como uma abordagem de auto-organização e discutiu-se que situações são mais indicadas a cada abordagem.

Terceiro, foi introduzida uma abordagem à modelação probabilística de sistemas distribuídos de robôs controlados por CAIs. Esta abordagem segue uma metodologia de modelação em múltiplos níveis, focando-se principalmente no nível macroscópico e usando o CAI como ponto de partida para um modelo probabilístico baseado em Redes de Petri Generalizadas e Estocásticas (RPGE). Estes modelos de RPGE foram aumentados com uma nova secção contendo informação sobre o estado do ambiente, onde informação não presente no CAI pode ser representada. Aplicou-se a abordagem de modelação a dois dos estudos de caso propostos, usando dados gerados por simulações submicroscópicas e estimativas obtidas a partir das propriedades físicas do ambiente como dados de entrada para os modelos de RPGE. Mostrou-se que esta metodologia é capaz de obter resultados coincidentes em diferentes níveis de modelação – macroscópico, microscópico e submicroscópico – e em experiências com robôs reais.

Palavras-Chave: Robótica Institucionalista, Robótica Distribuída, Controlo Robótico, Coordenação de Sistemas de Múltiplos Robôs, Modelação em Múltiplos Níveis, Redes de Petri Executáveis, Redes de Petri Generalizadas e Estocásticas, Economia Institucionalista, Simulação Realística de Robôs, Experimentação com Robôs Reais

Acknowledgments

After applying to a PhD student position I was lucky to be accepted in the IST/EPFL Joint Doctoral Initiative. This gave me the opportunity to split the time of my studies in half, spending one part in my home country and another in Switzerland. The first acknowledgment of this thesis must also be split in two parts, to both my advisors Prof. Pedro Lima and Prof. Alcherio Martinoli, for the constant support, guidance, patience, and confidence, they gave me during the time of my studies. The scientific knowledge and reasoning both of them have transmitted me is something for which I am extremely grateful. This thesis is finished due in a large part to their simple, direct and incisive comments.

I must also thank Dr. Porfírio Silva, who proposed the Institutional Robotics approach and is consequently behind the biggest motivation for this work. His enthusiastic embrace of a research area outside of his original comfort zone has motivated me to do exactly the same.

I must also thank my colleagues for the opportunity to work along fine researchers and fine individuals. To my colleagues at ISR, I thank for giving me my first taste of live in a research lab. To my colleagues at DISAL, I thank for transforming a lab in a home away from home. To all, I thank for the great discussions we had, some even about research, and I apologize for the times I annoyed you all on purpose. Thank you especially to Bruno and Chris for proof-reading parts of this manuscript and Aamir for helping with the delivery process.

To all my friends in Lisbon, Almeirim and Lausanne, some of which are also in the set of the colleagues mentioned above, thank you for the good times we spent and continue to spend together. Thank you for the bad times also. Thank you for making my live interesting and for being there whenever I needed.

To all my family, thank you. Mãe e Pai, outra vez sem palavras. Cheguei ao fim. Obrigado. To my grandparents, I wish one day I can give someone the support and love you give me. Sorry about the tattoos. To Clarisse, your smile shines the brightest ray of sunshine on my mind, it was well worth the wait to have a little sister. To Isabel

and Emanuel, uncles, aunts, cousins, second cousins and all the rest of you, thank you for being a family and making me feel part of the family even when away in a different country. I hope you will continue to do so.

And to Joana, with whom sharing a life is always worth the struggle of being separated in different countries. For whom I would go through it all over again. Thank you. This part is almost over, I can't wait for the next one.

(Anti-climactically) I would like to acknowledge that this work was supported in part by grants from FCT - Fundação Ciência e Tecnologia (ref. SFRH/BD/33671/2009) and ISR - Institute for Systems and Robotics (project ref. PEST3OE/EEI/LA0009/2011–ISR/IST) and by DISAL through the MOnarCH project (ref. FP7-ICT-2011-9-601033).

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Robots are increasingly becoming a part of our daily lives. While not every household has a robotic vacuum cleaner, and not every workplace has a Baxter[1] robot working amongst its employees, nowadays, the idea that robots can assist in everyday activities is not met with doubt or suspicion. Further, there is confidence that the pervasiveness of robotics in our lives will only increase with time.

Multi-robot systems are not as prevalent. At a high level, they present the same challenges one faces when working with their single-robot counterparts, including localization, navigation, sensing and actuation. Multi-robot systems additionally bring all the advantages and complexity associated with coordination. Once we are able to manage this complexity, multi-robot systems present enormous potential benefits over single robots, particularly in terms of redundancy and scalability. Thus the feasibility of such systems hinges on robots interacting with one another (and with humans) in a cooperative manner.

Current research aims to move the typically highly constrained applications of single- and multi-robot systems from the research lab to the real world. Examples of such efforts are the recently approved FP7-ICT[2] projects STRANDS[3] and MOnarCH[4]. In both projects, robotic systems (single-robot in STRANDS, multi-robot in MOnarCH) have to carry out tasks in human-populated environments and be able to sustain themselves for long periods of time with only rare interventions by system designers. Recently approved FP7-ICT coordination actions EURATHLON[5] and RoCKIn[6] take this goal further by creating robotic competitions in real-world scenarios where research efforts can be analyzed, compared,

[1] More info at http://www.rethinkrobotics.com/products/baxter/.
[2] More info at http://cordis.europa.eu/fp7/ict/.
[3] More info at http://strands-project.eu/.
[4] More info at http://monarch-fp7.eu/.
[5] More info at http://www.eurathlon.eu/.
[6] More info at http://rockinrobotchallenge.eu/.

and transmitted to industry partners who are responsible for their transition to the real world.

As we know, this transition from constrained laboratory environments to real-world environments is not a trivial step. Consider such real-world environments populated with human actors, even without any robotic systems present. For human actors carrying out any task, the need for coordination with other actors is ubiquitous. Simply traveling from point A to point B requires coordinating with others, be it on foot (using common sense rules to avoid colliding with others), by car (using a more elaborate set of rules, i.e., the "road code"), or any other mode of transport. More importantly, most tasks in our daily lives involve the trade of services or goods for money. This requires not only coordinating with others in order to physically carry out such a trade but also the underlying acceptance that the money being received can then be accepted by some other actors. The collective acceptance of such an idea is in itself a form of society-wide coordination.

For robotic systems to be truly immersed in real-world environments this type of coordination must be considered. Robots need to consider complex social interactions with multiple anonymous robots and multiple anonymous human actors, where this anonymity reflects the fact that interacting agents might never have met before. Such interactions may also be characterized by the uncoupling of time and space, meaning that interacting agents do not have to coexist at the same time or place for the interaction to occur.

Given the properties of such complex social interactions, we choose to focus our work on the study of distributed robotic systems. Such multi-robot systems are characterized by the properties of control algorithms being decentralized and all robots being equal with respect to the coordination mechanism. The properties of space and time uncoupling are more easily captured by a decentralized approach than by a centralized one. The distributed approach to control also allows that several applications of distributed robotic systems, in particular swarm robotics, consider large numbers of interacting agents. To allow for a scalable increase in the number of agents, anonymity must usually be taken into account.

Obviously, our goal is not to obtain an absolute methodology for specifying social interactions that accounts for all possible instances of coordination problems. As noted in [Durfee, 2004] (about multi-agent systems), "It does not seem possible to devise a coordination strategy that always works well under all circumstances; if such a strategy existed, our human societies could adopt it and replace the myriad coordination constructs we employ, like corporations, governments, markets, teams, committees, professional societies, mailing groups, etc." However, we can try to formalize rules specifying social interactions for specific tasks in a way that resembles the organizational aspects of human society. Our intuition is that by doing so we will be able to consider complex social interactions within distributed robotic systems and ease the effort of their transition to real-world environments populated with human actors.

To do so, we follow the *institutional robotics* approach [Silva and Lima, 2007] to the coordination of multi-robot systems. This approach takes inspiration from social sciences, namely from *institutional economics*, and aims to provide a comprehensive strategy for specifying complex social interactions among a team of robots and possibly between a team of robots and human actors. In this work, we will formalize some concepts associated with the institutional robotics approach in order to allow for their implementation in real robotic systems.

This chapter is organized as follows. In Section 1.1 we discuss the motivation behind this work. Section 1.2 presents the objectives of the thesis, while Section 1.3 gives an overview on the organization of the remaining chapters of this work. Finally, Section 1.4 presents the major contributions of the work and lists the relevant published work.

1.1 Motivation

The more general motivation behind this work is named in the title of this manuscript: to advance the level of social interactions among robots. As discussed above, complex coordination problems require coordinating agents to maintain complex social interactions. These issues are well-studied by several fields in the social sciences, particularly in economics. Although an all-encompassing coordination strategy is not available, several coordination constructs (organizations, teams, norms, social roles, etc.) can be captured by a single concept: *institutions*. This is the central concept in institutional economics. However, due in part to making use of ideas and data from other social sciences (e.g. psychology, sociology, and anthropology) and in part to not being concerned with prescribing policies, institutional economics lacks some formality (in a mathematical sense) at its core.

Institutions are also the central concept in institutional robotics, where the coordination strategy for a given multi-robot system carrying out a given task (or more than one) is considered to be given by a network of institutions. Due to its inspiration from social sciences, the institutional robotics approach is prescribed in [Silva and Lima, 2007] as a set of guidelines for devising the coordination strategy. In order to actually implement such an approach in distributed robotic systems a higher degree of formality in the methods is considered necessary.

The main motivation for this work is to advance institutional robotics by casting and expanding the previously proposed theoretical framework to the distributed robotic systems reality. To do so, we need to provide formal methods to coordinate robots, in particular formalizing the concept of institution to allow its inclusion in robot controllers. While expressing the concept in a more formal version, we must guarantee that the core guidelines of the institutional robotics approach remain valid. This is not a trivial task when dealing with concepts rooted in the study of social sciences.

Due to the size and scope of application of the type of problems we are interested in, not only are control and coordination methods necessary, but we are also in need of formal modeling methods. Such methods allow us to capture at a macroscopic, probabilistic level the overall behavior of teams of robots endowed with controllers realizing the institutional approach, and to identify in which situations coordination among robots is achievable. The capability to predict the behavior of large distributed robotic systems is of major importance in this field, due to the stochastic, dynamic, and non-linear, nature of such behavior and the inherent experimental difficulties of working with a large set of robots. Based on this motivation we develop formal modeling methods for the institutional robotics approach.

Up to the start of this work, the institutional robotics approach had been proposed only theoretically and remained unimplemented, in terms of real or simulated robotic experiments. Another source of motivation is to move towards experimental implementation. Experimental validation combined with formal modeling methods will allow us to study the impact of the approach. In this work we will present several case studies designed to focus on different aspects of the institutional robotics approach and to validate our approaches to coordination, control, and modeling.

1.2 Objectives

Following our motivation, the main three objectives for this work can be described by the following items:

1. **formalizing** institutional robotics' concepts from a computer science perspective, leading to coordination and control methods for distributed robotic systems where complex social interactions are taken into account;

2. **implementing** and validating concepts from institutional robotics, both in simulated and real robots, for laboratory scenarios designed to put forward relevant questions about the institutional approach, comparing the results against other existing approaches;

3. **modeling**, using formal mathematical methods, the distributed robotic systems designed and implemented under the institutional robotics approach, providing sound tools for design evaluation and performance analysis (both qualitatively and quantitatively) and further comparison with other approaches.

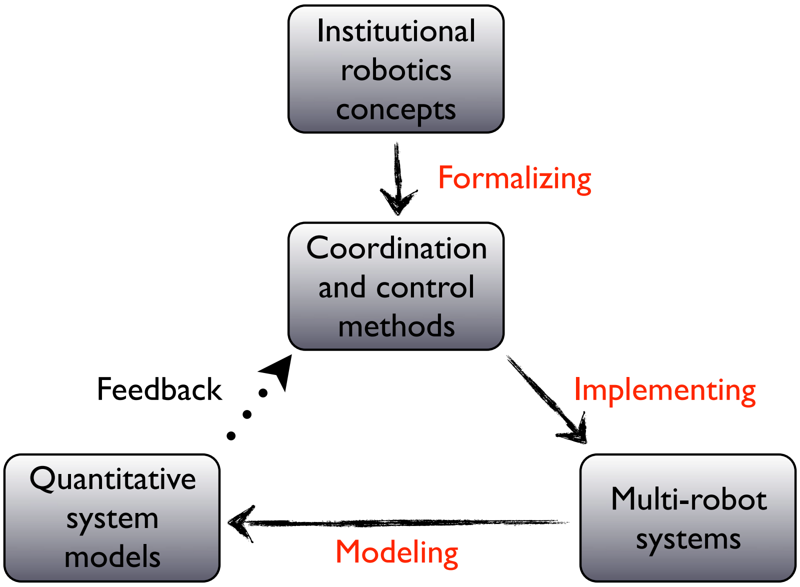


Figure 1.1: Thesis objectives (formalizing, implementing, modeling), their end results and relationships.

These objectives are highly intertwined, with insights from the work in each objective being relevant for the other two. Fig. 1.1 captures some of those relationships.

The formalization of institutional robotics' concepts is our main objective for this work. Central to our formalization will be the concept of *institution*, together with the concepts of *coordination artifact* and *institutional environment*. Institutions will be formalized as coordination artifacts using a specialized extension of a formal modeling tool called Petri nets. The use of this tool will allow us to represent institutions in robots and use those institutions as a control method. The composition of several institutions will lead to an *institutional agent controller* where the institutional environment of each robot is represented. Such a controller will be replicated in all the nodes of a distributed robotic system, thus replicating the institutional environment for each robot and providing the necessary coordination for the system.

The institutional agent controller is also used in our modeling efforts, where we will borrow some concepts from a multi-level probabilistic modeling methodology previously established for swarm robotic systems. This methodology takes into account the individual robot controller in order to generate an aggregated macroscopic representation of the dynamics of the whole system. Such models will allow for quantitative performance analysis of distributed robotic systems under the institutional approach. It is our intuition that if formal control and modeling methods can be bridged, this will represent an advantage for designers of distributed robotic systems.

Several case studies are implemented. A simple swarm robotics case study, in which both control and modeling methods have been successfully applied, will be used to validate both the control methods based on the formalization of institutions and our approach to probabilistic modeling under the institutional approach. Two other case studies of a higher degree of complexity are implemented, in which institutional robotics' concepts will be studied in more detail.

1.3 Organization

This manuscript is divided in three distinct parts as well as an introductory chapter and a concluding chapter. It can be described as follows.

- **Chapter 1** introduces the work, describing the motivation and objectives.
- **Part I** provides the starting point from which the work described in the following parts is achieved.
  - **Chapter 2** presents the robotic platform and computational tools used for experimentation and validation.
  - **Chapter 3** describes the institutional robotics approach and its inspiration in social sciences, namely institutional economics. We discuss why other approaches to the coordination of distributed robotic systems do not satisfy our motivation of considering complex social interactions in such systems. From this starting point we describe institutional economics and how it can serve as a good inspiration for a coordination mechanism for distributed robotic systems.
- **Part II** presents the main methodologies developed in this work, for control and modeling of distributed robotic systems under the institutional robotics approach.
  - **Chapter 4** formalizes of the concepts of institution and institutional agent controller. We introduce the tool of executable Petri nets, which takes into account robot actions and sensor readings to allow the design of controllers for robots. We present the formalization of institutions using this tool and discuss how different institutions can be composed into an institutional agent controller that can be executed by individual robots.
  - **Chapter 5** introduces our approach to the probabilistic modeling of distributed robotic systems under the institutional robotics approach. We introduce the tool of Generalized Stochastic Petri Nets and discuss how the Petri

net structure of the institutional agent controllers can be used as the starting point in a multi-level modeling methodology. We discuss how to extend this initial model in order to take into account information from the physical environment.

  - **Chapter 6** presents the validation of the control and modeling methodologies described in the two previous chapters using the *wireless connected swarm case study* (previously introduced in [Nembrini et al., 2002]). We implement two controllers for the case study: the original version using a finite state automata and an institutional agent controller. We compare results from both controllers in simulation. These simulations are validated by implementing the case study with a swarm of as many as 40 real, resource-constrained robots. Using the institutional agent controller designed, we construct a probabilistic macroscopic model for the overall state distribution of the system.

- **Part III** contains the larger part of the implementation effort for the institutional robotics approach, describing two distinct case studies.

  - **Chapter 7** demonstrates how concepts from institutional robotics can be applied in a robotics task, focusing on one specific form of institution, the institutional role. To do so, we introduce the *corridor case study*, where robots must coordinate when navigating through the environment in order to accomplish the task. We compare results of the institutional approach with a self-organized approach and identify which conditions are more suitable to each approach. We also present a modified version of the case study in which robots adapt their controller to accommodate changes in the environment.

  - **Chapter 8** discusses the impact of considering institutions in heterogenous distributed robotic systems that are not energetically sustainable without an appropriate coordination strategy. We introduce the *piece assembly case study*, where the main mission of the robots in to gather the necessary components to assemble certain types of pieces. Robots spend energy while carrying out the task but obtain energy rewards when the team goal is accomplished. The sustainability of the system is related to its ability to obtain sufficient energy rewards. Different types of the robots in the team make different decisions on a social dilemma concerning the priority of individual over collective goals. We compare a purely decentralized and an institutional approach and study how these approaches impact the sustainability and efficiency of the team. We also apply our modeling methodology in order to obtain an *a priori* probabilistic model of the system.

- **Chapter 9** closes this work with concluding remarks and presents directions for future work with the institutional robotics approach.

1.4 Contributions & Publications

The contributions of this work to the advance of the state of the art can be summarized as follows:

- introduction of **Executable Petri Nets**, an extension to the Petri net tool;
- development of a methodology for distributed robotics systems based on the **formalization** of the concept of institution and institutional agent controller;
- adaptation and application of a probabilistic **modeling** methodology to the institutional robotics approach;
- **validation** of control and modeling methodologies in real and simulated robots;
- **implementation** of the formalized institutional robotics concepts in three distinct case studies, thereby advancing the validation effort of the institutional robotics approach.

These contributions can be found in the following publications:

- [Pereira et al., 2010], Pereira, J. N., Christensen, A. L., Silva, P., and Lima, P. U. (2010). Coordination Through Institutional Roles in Robot Collectives (extended abstract). 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
- [Pereira et al., 2011], Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2011). Formalizing Institutions as Executable Petri Nets for Distributed Robotic Systems. In Lenaerts, T., Giacobini, M., Bersini, H., Bourgine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011*, pages 646–653, Paris, France. MIT Press
- [Pereira et al., 2013], Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2013). An Experimental Study in Wireless Connectivity Maintenance Using up to 40 Robots Coordinated by an Institutional Robotics Approach (accepted). In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*
- [Pereira et al., 2014a], Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2014a). Formalization, Implementation, and Modeling of Institutional Controllers for Distributed Robotic Systems (in print). *Artificial Life*, 20(1)
- [Silva et al., 2013], Silva, P., Pereira, J. N., and Lima, P. U. (2013). Institutional Robotics. Institutions for social robots. (submitted). *International Journal of Social Robotics*

- [Pereira et al., 2014b], Pereira, J. N., Tarapore, D., Silva, P., Martinoli, A., and Lima, P. U. (2014b). Considering Institutions in Unsustainable Robotic Systems (in preparation). *Autonomous Agents and Multi-Agent Systems*

Part I

Preliminaries

Chapter 2

Materials and Methods

In this chapter we describe the robotic, simulation, and computational tools used for experimentation and validation in the following chapters. Despite being very useful to this purpose, the tools described in no way impose restrictions on the methods of control and modeling to be presented. These methods are independent of the chosen robotic platform or computational tools used.

This chapter is organized as follows. In Section 2.1 we describe the robotic platform used, the *e-puck* robot. Section 2.2 describes the robotic simulators used. Finally, Section 2.3 describes other computational tools used, including Petri net tools and video processing software.

2.1 E-puck

The robotic platform used in this work is the *e-puck* robot[1] [Mondada et al., 2009], shown in Fig. 2.1. The *e-pucks* are small (7 cm diameter) wheeled robots designed at EPFL for use in educational and research experiments. The robots are simple, relatively inexpensive and robust, which makes them suitable for experiments in collective robotics. Each *e-puck* is equipped with a variety of sensors. In our experiments, we use the *e-pucks'* proximity sensors, differential drive system, bluetooth communication, and camera.

In order to endow the robots with scalable wireless communication capabilities, we use a radio communication module developed at DISAL [Cianci et al., 2006]. Fig. 2.1-(b) shows an *e-puck* robot equipped with this module, stacked between the main board and the speaker board. The radio communication module is Zig-Bee compliant and uses

[1]See http://www.e-puck.org.

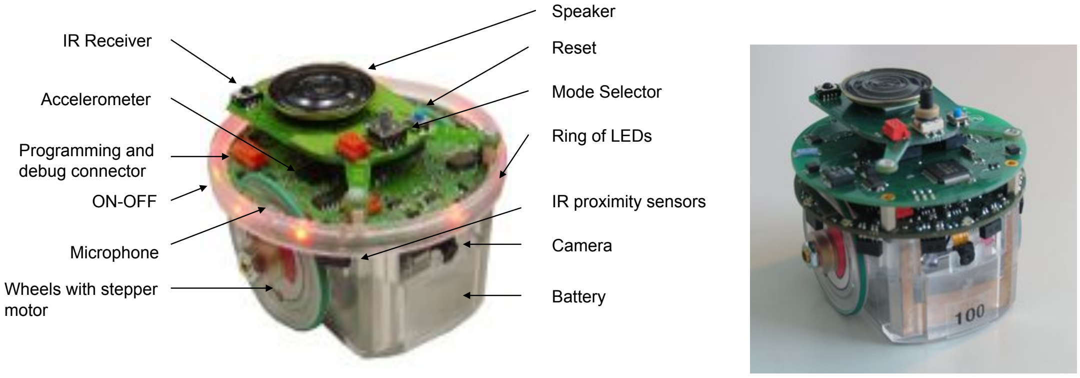


Figure 2.1: (a) The *e-puck* robot. (b) *E-puck* equipped with radio communication module

TinyOS [Levis et al., 2004]. A bounded communication range is obtained using software controllable power emission and a hardware attenuator.

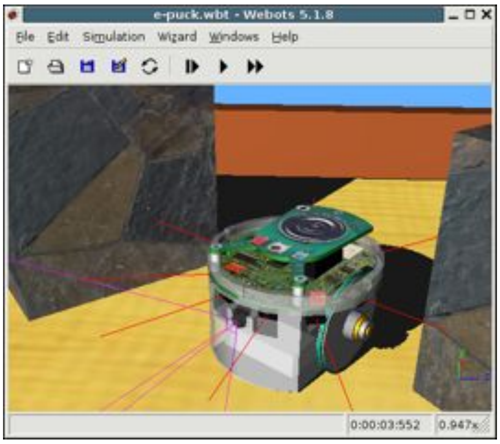
2.2 Simulators

In the case studies described in Chapters 6 and 7, we used *Webots*[2] [Michel, 2004], a submicroscopic, flexible, 3D realistic simulator, and considered kinematic models of the *e-puck* robot. A screenshot from a simulation with the *e-puck* model is shown in Fig. 2.2-(a).

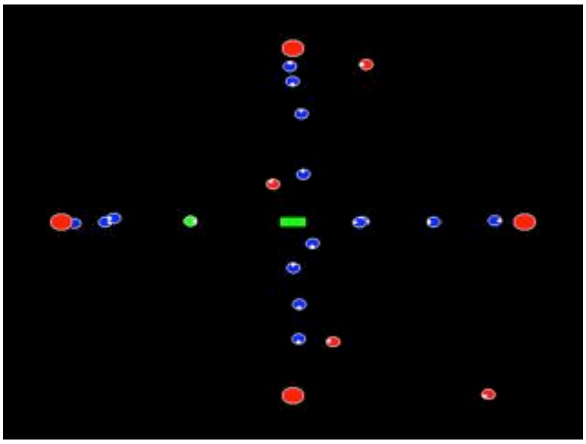
To replicate the use of the radio communication module described in the previous section, communication between *e-pucks* is simulated realistically using the network simulation engine OMNeT++ [Varga, 2002] as a plugin for *Webots*. The OMNet++ engine handles channel coding, noise, fading signal propagation, as well as a non-circular communication footprint. This plugin was used when comparing results from simulation and reality in Chapter 6.

The case study described in Chapter 8 was implemented in the *twodeepuck* simulator, a microscopic, stochastic, spatial, discrete-time simulator for *e-puck* robots. The *twodeepuck* is a modification of the *twodee* simulator [Christensen, 2005] designed for the *Swarm-bot* [Mondada et al., 2004]. The *twodeepuck* simulator was the simulator used for the project “From Bio-Inspired to Institutional-Inspired Collective Robotics”[3] (BioInstBots).

[2]See http://www.cyberbotics.com.
[3]FCT-sponsored project (Ref: PTDC/EEA-CRO/104658/2008) - http://mediawiki.isr.ist.utl.pt/wiki/From_Bio-Inspired_to_Institutional-Inspired_Collective_Robotics



(a)



(b)

Figure 2.2: (a) The *e-puck* robot module in a *Webots* simulation. (b) *twodeepuck* simulation screenshot (small circles represent agents).

A screenshot from a simulation is shown in Fig. 2.2-(b).

2.3 Other Computational Tools

During the experiments with real *e-pucks* described in Chapter 6, we recorded videos of the arena using an overhead camera and the *SwisTrack*[4] software [Lochmatter et al., 2008]. *SwisTrack* is a software for tracking robots, humans, animals and objects using a camera or a recorded video as input source. We processed the videos offline, using *SwisTrack* to perform background subtractions and blob detection, in order to extract and store the position of each robot in each frame.

For edition and analysis of Petri nets and Generalized Stochastic Petri nets, we used the software *PIPE2*[5] (Platform Independent Petri Nets Editor) [Bonet et al., 2007, Dingle et al., 2009]. *PIPE2* is an open-source tool that supports the design and analysis of Generalized Stochastic Petri net models. *PIPE2*'s extensible design enables developers to add functionality via pluggable analysis modules. It also acts as a front-end for a parallel and distributed performance evaluation environment. However, some more complex analysis dealing with Generalized Stochastic Petri nets with marking-dependent firing delays was performed using *TimeNET*. *TimeNET*[6] (Timed Net Evaluation Tool) [Zimmermann and

[4] See http://en.wikibooks.org/wiki/SwisTrack.
[5] See http://pipe2.sourceforge.net/.
[6] See http://www.tu-ilmenau.de/sse/timenet/.

Freiheit, 2000, Zimmermann and Knoke, 2006], is a software tool for the modeling and analysis of stochastic Petri nets with non-exponentially distributed firing times.

Summary

In this chapter we describe the robotic, simulation, and computational tools used for experimentation and validation in the following chapters. We describe our robotic platform, the *e-puck* robot, and a radio communication module used to achieve local communication between robots. We introduce the two simulators used in different case studies. In conclusion, we introduce other computational tools that were used in the work to be presented in the next chapters.

Chapter 3

Institutional Robotics

Multi-robot systems are nowadays an important area of research within the broader field of robotics. An increase in team size from the single robot alternative may lead, in particular applications, to benefits for the system, not only in the performance of its specific task, 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 a single robot, e.g., due to their physical properties. However, to draw these rewards it is not enough to simply add robots to the team. Cooperative behavior [Cao et al., 1997] has to be present, and to achieve it interactions among robots must be coordinated in some way. In recent years, researchers have proposed various methods for coordination of multi-robot systems. In this thesis, we focus only on distributed robotic systems, characterized by the properties of being decentralized and all robots being equal with respect to the coordination mechanism.

A common point between several existing approaches to the coordination of distributed robotic systems is the low complexity of the social interactions considered. The *institutional robotics* approach [Silva, 2007, Silva and Lima, 2007] has been introduced to coordinate multi-robot systems, in particular aiming to provide a comprehensive strategy for specifying complex social interactions among robots of a team and possibly between a team of robots and human actors. To do so, this approach draws inspiration from social sciences, namely from institutional economics [Hodgson, 2000]. Our intuition is that by replicating some aspects of the organization of human societies not only will we be able to increase the complexity of social interactions within distributed robotic systems but also we will ease the insertion of such systems into real world scenarios populated with human actors.

This chapter is organized as follows. In Sections 3.1 and 3.2 we discuss two other approaches to the coordination of distributed robotic systems, namely self-organized systems and market-based systems. Section 3.3 describes other instances where concepts from social sciences have been applied to multi-agent systems. In Section 3.4 we introduce our main inspiration for institutional robotics, institutional economics, and in Section 3.5 we discuss new institutional economics, a particular branch of institutional economics with less imposing assumptions. Section 3.6 is dedicated to the definition of the concept of institution from an ontological perspective. In Section 3.7 we present the institutional robotics approach followed in the thesis.

3.1 Self-organized systems

Self-organization is a possible, scalable mechanism that has been proposed for the coordination of, often large, distributed robotic systems [Bonabeau et al., 1999, Beni, 2005, Sahin, 2005, Beckers et al., 1994, Krieger and Billeter, 2000, Mondada et al., 2005]. Self-organizing systems are characterized as being fully reactive and relying on local interactions (and possibly local, broadcast communication), both between robots and between robots and the environment, in order to achieve coordination. This leads to low hardware requirements for the individual robot platform, and consequently to the possibility of implementation on a large number of cheap, simple robots.

Self-organizing coordination algorithms reported up to date are extremely robust [Winfield and Nembrini, 2006] although not necessarily efficient [Pettit, 2003]. They also often underexploit the potentially high level of cognition and networking available at the individual robotic node. For instance complex distributed robotic systems might need indirect or mediated interaction (such as economic relations in a market with money), characterized by properties such as time uncoupling and space uncoupling (interacting agents neither have to be at the same place nor to coexist at the same time) [Weyns et al., 2005a]. Although some self-organizing systems can be said to account for mediated interaction (for instance pheromone-based foraging algorithms [Payton et al., 2001]), differentiation of that interaction as part of a larger set of (social) rules or as an interaction with the environment is not achievable without some degree of cognition. This differentiation is fundamental in obtaining complex cooperative behavior (as in the differentiation of a hundred dollar bill from a piece of paper is fundamental in economic relations).

Moreover, autonomy does not mean self-sufficiency, and any agent enjoys, at best, bounded autonomy (it depends on a resource or on other agents to achieve its own goals), with the possibility of different agents having different goals. This means that the design of truly social distributed robotic systems should take into account the objective social interactions arising from the combination (and dependencies) of goals of heterogenous agents [Conte and Castelfranchi, 1995]. Although it is possible to consider heterogenous agents in a self-organizing system [Dorigo et al., 2012, Ducatelle et al., 2010, Li et al., 2004], typically all agents strive towards a common collective goal. Therefore, self-organization is probably not the best guiding principle to deal with different degrees of autonomy enjoyed by different agents.

3.2 Market-based systems

Market-based multi-robot coordination [Dias et al., 2006] is a previous example of importing some economic views into robotics. Inspired by market mechanisms, researchers have proposed systems like MURDOCH [Gerkey and Mataric, 2002] and TraderBots [Dias et al., 2004] to achieve flexible allocation of subtasks using auctions between robots. In these systems, robots act as agents trying to maximize their individual profits. Every time a task is auctioned, robots must pay a price to obtain it. Once the task is completed, a payment is done to the robot who won the auction. Nevertheless, to accomplish the task, that robot has to expend some resources for which it must also pay a price. In these systems, tasks and resources are considered as commodities than can be compared in value and traded among robots. The underlying assumption is that with every robot trying to maximize its individual profit, team coordination and efficiency will be improved. A limitation of the market-based approach is that, despite some application to the allocation of roles [Vail and Veloso, 2003], the great majority of the work available only deals with task allocation, leaving other mechanisms (e.g., cooperative decision-making) out of the picture.

Market-based systems can be said to have an intentional model of cooperation [Parker, 1998], where different tasks have to be accomplished and robots cooperate explicitly, often through communication, to allocate correctly resources to tasks. Since our endeavor is to advance social interactions between robots, we believe that this intentional model of cooperation is also the more appropriate for our system. As Gerkey states in [Gerkey and Mataric, 2002], "if the robots are deliberately cooperating with each other, then, intuitively, humans can deliberately cooperate with them, which is a long-term research goal of multirobot research.". Nevertheless, we do not wish to take the assumptions of market-based approaches about the end result of the maximization of individual profit.

3.3 Inspiration from social sciences

Inspiration from social sciences other than economics has been very present in multi-agent systems research in recent years. This indicates an attempt not only to advance the understanding of how different social interactions amongst individuals shape the behavior of the whole system, but also to design better multi-agent systems building upon concepts well established for human societies. Several concepts have already been adopted in the framework of multi-agent systems (e.g., norms [Hexmoor et al., 2006], conflicts [Malsch and Weiß, 2001], trust [Sabater and Sierra, 2005], reputation [Hahn et al., 2007], individual rights and argumentation [Alonso, 2004]). However, even if such a variety of approaches may prove useful in terms of scalability [Durfee, 2004], it is often too fragmentary. We need a unifying concept to give the whole some consistency.

Also, there is a need to consider explicitly the design of coordination devices, since it has been shown that even within highly formalized deliberative mechanisms (e.g., courts, selection committees), or where purposive groups have to take coherent decisions over a period of time, collective reason does not necessarily emerge from individual's reason alone [Caldas, 2001]. A similar phenomenon can also be observed in multi-agent systems using reinforcement learning so that agents learn individually a sequence of actions that carried out jointly would lead to a predetermined global objective. Agents might maximize individual payoff but, due to working at cross-purposes, the global objective is not achieved [Tumer et al., 2002]. These studies provide some intuition that probably not all problems requiring a global coordination strategy can be solved by considering the maximization of individual profit as the sole motivation for agents.

3.4 Institutional Economics

In an effort to give an answer to the need for a unifying concept from social sciences and the need for inclusion of coordination devices, we look for inspiration in institutional economics, which is by itself a multi-disciplinary approach. It is a fundamentally different approach from neoclassical theory, the current mainstream trend of economics and inspiration for market-based systems.

In [Hodgson, 1988], Geoffrey Hodgson asserts that classical neoclassical theory relies on three assumptions he deems incorrect:

- all economics agents are considered to follow a rational, utility-maximizing behavior;
- absence of chronic information problems - uncertainty about the future, lack of information about world structure and parameters, divergence on the reasoning of economical agents about individual phenomena;
- theoretical preference for stable equilibrium states and lack of regard for historical transformation processes and their dynamics.

Based on his disagreement with these assumptions Hodgson proceeds to derive critiques on three points of the neoclassical approach [Hodgson, 1988]:

1. critique of methodological individualism;
2. critique of the maximization hypothesis;
3. critique of the rationalist concept of action.

Methodological individualism postulates that any proper explanation of social phenomena must be formulated in terms of individuals, since all actions are executed by individuals and collectives are not more than the actions of their individual members. Individual action is connected to collective phenomena through successive and hierarchical compositions. The institutional approach rejects that individual behavior might be the only possible explanation for such phenomena. Institutions influence individual behavior, enabling collective behavior that is more than the sum of its parts.

Hodgson then presents a critique of the *maximization hypothesis*. Rational economic agents, as considered by the neoclassical approach, choose actions by maximizing a certain value expressed as a single variable (e.g., utility or profit). Such maximization is performed taking into account all the relevant information available in the world and depends on the assumption that individuals have a transitive and irreflexive ordering of their preferences. Hodgson disagrees with such approach, rejecting that: there is a single value to be maximized; it is possible to obtain all relevant information; and it is possible to obtain such an ordering on the preferences of individuals. Note that, in neoclassical economic theory, the notions of maximizing rationality and equilibrium are deeply connected. Equilibrium is reached when all economic agents reach the maximum of their utility functions, obtained following a rational and global (w.r.t. information) calculus.

Finally, there is the critique of the *rationalist concept of action*. Not all relevant actions in economy are governed by rational calculus. For instance, there are unconscious and subconscious mental processes that influence action. The effect of publicity is a good example of such processes. Hodgson considers that, for agents with bounded rationality (as human being are), it is important that not all processes are conscious and deliberative. This allows agents to focus their rational capabilities on actions that effectively need it. Some of those processes take the form of routines and habits. Beliefs, attitudes, and moral values, also shape our interpretation of reality in a non-deliberative way. All these processes are social institutions. Such institutions not only spare the rational capabilities of agents but also reduce their uncertainty, by allowing them to predict the behavior of other agents who also consider these institutions.

In [Hodgson, 2000], Hodgson refines a description of institutional economics based on the 1919 paper of Walton Hamilton [Hamilton, 1919]. This description is captured in the following five points [Hodgson, 2000].

1. Although institutional economists are keen to give their theories practical relevance,

institutionalism itself is not defined in terms of any policy proposals.

2. Institutionalism makes extensive use of ideas and data from other disciplines such as psychology, sociology, and anthropology, in order to develop a richer analysis of institutions and of human behavior.

3. Institutions are the key elements of any economy, and thus a major task for economists is to study institutions and the processes of institutional conservation, innovation and change.

4. The economy is an open and evolving system, situated in a natural environment, affected by technological changes, and embedded in a broader set of social, cultural, political, and power relationships.

5. The notion of individual agents as utility-maximizing is regarded as inadequate or erroneous. Institutionalism does not take the individual as given. Individuals are affected by their institutional and cultural situations. Hence individuals do not simply (intentionally or unintentionally) create institutions. Through "reconstitutive downward causation" institutions affect individuals in fundamental ways.

Note that the institutional approach is characterized also by the rejection of determinism. 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.

Of the five points above, two are of critical importance to the institutional robotics approach. Point five dismisses a complete reliance on the maximization hypothesis in any institutional approach. Although agents may take deliberative actions, choosing a certain decision based on some utility measure, these are not the only possible actions. Moreover, even deliberative actions cannot be removed of the institutional environment in which they are executed. Institutions are created by agents, but they also affect how agents choose a given course of action. The institutional environment is critical for the behavior of agents with bounded rationality, category in which the robots we will consider belong to.

This can be seen in the experiments of [Caldas, 2001] described in Section 3.3, where even within highly formalized deliberative mechanisms, collective reasoning cannot be disassociated from the institutional context. The structure of a situation can be the main factor causing the observed actions of agents, as opposed to considering only their internal motivations or capabilities. In a series of computational economics experiments [Gode and Sunder, 1993, Gode and Spear, 2004, Bosch-Domènech and Sunder, 2000], authors have shown that situations previously explained, under the maximization hypothesis and the rationalist concept of action, as a result of individual rational behaviour, can be explained by the institutional setup itself. Specifically, they have shown that Pareto efficient outcomes are achieved within double auction contexts, a very specific market institution, by "zero-intelligence" traders, a trivial type of agent without high level intelligence that simply submits random bids (under some simple imposed constraints).

Point three states the importance of institutions in economy. Institutions will be the unifying concept proposed by institutional robotics to advance social interactions among robots, taking the role of coordination devices. In the following sections we will look further into this concept. An important property of institutions is that they allow mediated interaction between agents, which [Weyns et al., 2005a] considers characterized by name uncoupling (interacting agents do not have to know one another explicitly), space uncoupling and time uncoupling (they do not have neither to be at the same place nor to coexist at the same time).

Money is a classical example of the power of institutions in providing the means for mediated interaction [Coase, 1992]: "Adam Smith pointed out the hindrances to commerce that would arise in an economic system in which there was a division of labor but in which all exchange had to take the form of barter. (...) A person wishing to buy something in a barter system has to find someone who has this product for sale but who also wants some of the goods possessed by the potential buyer. Similarly, a person wishing to sell something has to find someone who both wants what he has to offer and also possesses something that the potential seller wants." The use of money overcomes this difficulty allowing mediated or indirect interaction.

Incompleteness in terms of information and heterogeneity of individual agents are some other concepts of institutional economics that will be explored in the institutional robotics approach.

3.5 New Institutional Economics

Over more recent years, an approach that deviates both from the mainstream neo-classical theory and from the alternative view of institutional economics has come to light. The "new institutional economic" approach recognizes the importance of modeling institutions in capturing the correct behavior of economic agents that live in a real world. On the other hand, it does not reject all the principles of neo-classical thought, especially the notion of agent as utility-maximizing, as did its "old" institutional economics counterpart [North, 1993, Hodgson, 1998]. Despite of that, new institutional economics does view the agent as having incomplete information and bounded rationality. It is to face this uncertainty that institutions appear, as a way to control the costs inherent to any economic exchange (costs of obtaining information about a particular good, trade, contract, etc.) [Menard and Shirley, 2005].

Another important focus of new institutional economics is on how institutions emerge,

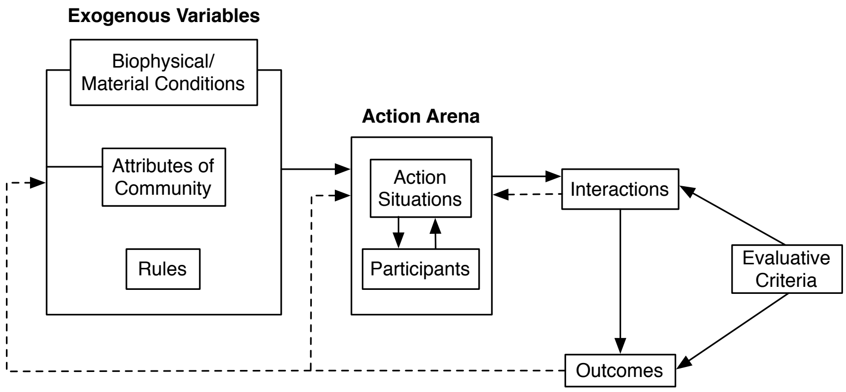


Figure 3.1: A framework for institutional analysis. Source: [Ostrom, 2005], page 15.

sustain and evolve, and what rules do these mechanisms of institutional growth follow (if any). In order to study these questions, Elinor Ostrom developed the institutional analysis and development framework (IAD) [Ostrom et al., 1994, Ostrom, 2005]. This framework studies the arena where interactions occur, the rules used by agents in their individual behaviors, and the attributes of an environment that structures and is structured by interactions among agents present in that environment (Fig. 3.1).

Ostrom also proposes a formal "grammar" of institutions [Crawford and Ostrom, 1995, Ostrom, 2005], in which institutional statements about situations where humans interact can be composed using elements as deontic operators, attributes and conditions. We claim that a more formal definition for institutions, such as the IAD framework, can help us specify more accurate models of the environment and interactions in our multi-robot system. In fact, elements of this grammar of institutions will have a direct impact in capturing the institutional environment while designing controllers for the robots.

3.6 What are Institutions?

In the introductory essay to an anthology of new institutional economics contributions [Menard and Shirley, 2005], Ménard and Shirley say that "institutions are the written and unwritten rules, norms and constraints that humans devise to reduce uncertainty and control their environment", including constitutions, laws, unwritten codes of conduct, norms of behaviour, and beliefs. While this statement gives some insight to the possible impact of considering institutions in multi-robot systems, it is still far away from a formal definition.

The question about exactly what are institutions has a misleading simplicity to it. It is a question that has been tackled by numerous social scientists, both from economics (for instance, in Hodgson's "What are institutions?" [Hodgson, 2006]) and other social sciences (for instance, in Searle's "What is an institution?" [Searle, 2005]). However, there is not one consensual definition. In this section we follow the work of Searle in establishing a fundamental ontology of institutions.

Instead of directly answering the referred question, Searle [Searle, 2005, Searle, 2006] chooses to analyze what are *institutional facts* and in what way they differ from other types of facts, which he categorizes as *brute facts* and *social facts*. *Brute facts* can be said to be observer-independent, objective facts, meaning they do not need to be experienced by an agent in order to exist. Searle's analysis of institutional facts, and ultimately of institutions, relies on three primitive elements necessary to explain social and institutional reality.

Collective Intentionality

Collective intentionality is a capacity of humans (and some other species) to engage in cooperative behaviour and sharing of attitudes with other humans (or species members). "Intentionality", in the philosophical sense of the word, describes the feature of mind by which mental states are directed at (or are about) objects and states of affairs in the world. Intentionality is not about the intentions of agents. Rather, their beliefs, hopes, desires (and indeed intentions) can in this technical sense be said to be intentional. Besides individual intentionality, we are also capable of collective intentionality. Collective intentionality can take the form of intentional collective action (playing a violin part as part of the orchestra playing the symphony) or other forms, like a collective belief (a church congregation reciting a prayer is expressing a common belief that is an identity mark of the community). In Searle's terms, a *social fact* is any fact involving collective intentionality of two or more agents.

Status functions

Humans, and some animals, have the capacity to collectively assign functions to objects or persons. If an individual can use a stump as a chair, a group can use a log as a bench. Here, the assignment of function is supported on physical features of the objects. Moreover, humans have the capacity to assign functions to objects or persons, where the function cannot be performed just in virtue of their physical features, but rather in virtue of the fact that there is a collective assignment of a certain status. In this case we speak of status functions. For instance, money, as a function, does not depend on the material chosen for banknotes or coins, but depends of the collective agreement to use such physical support to represent wealth and aid trade. Institutions and institutional facts are created and exist thanks to special acts of collective intentionality dealing with the collective assignment, recognition, and acceptance, of status functions.

The general form of the assignment of a status function ("constitutive rule") is "X counts as Y in context C". The X term identifies certain features of an object or person or state of affairs, and the Y term assigns a special status to that person, object, or state of affairs. Money is an institution in which a certain kind of piece of paper, produced under certain circumstances, is taken as currency and performs a function that can be described as general equivalent for exchange: "X (this piece of paper) counts as Y (a five euro bill) in context C (the Euro zone)". Marriage is an institution in which certain words (X), uttered by the right person in the correct circumstances (C), serve as the beginning of a certain kind of relationship between the people involved (Y), implying specific rights and duties. In any case, the Y term must name something more than the sheer physical features of the object named by the X term.

Deontic Powers

Status functions are vehicles of power in human society. We accept status functions and in so accepting, we accept a series of obligations, rights, responsibilities, duties, permissions, and so on. All these are deontic powers. For instance, the property of some land gives the right to build on it and the obligation to pay some taxes. In human societies, we have a set of deontic power relations. Obligations and permissions are reasons to perform certain actions, if we recognize the appropriate opportunities. Importantly, deontic power relationships provide reasons for action that are independent of desires. The recognition of land ownership by people travailing on a certain terrain gives them reasons for certain actions that are imposed by deontic power relationships and not by any personal desire, for instance, the obligation to leave as soon as possible or the responsibility to not litter.

With these three elements, Searle [Searle, 2005] defines institutional facts and institutions. *Institutional facts* can be said to be any fact that has the logical structure "X counts as Y in context C", where the Y term assigns a status function and this status function carries a deontology. An *institution* is any system of constitutive rules of the form "X counts as Y in context C". In closing, Searle provides a way to answer the question "is W (word) an institution?" [Searle, 2005] :

1. "Is W defined by a set of constitutive rules?
2. Do those rules determine status functions, which are in fact collectively recognized and accepted?
3. Are those status functions only performable in virtue of the collective recognition and acceptance, and not in virtue of the observer-independent features of the situation alone?
4. Do the status functions carry recognized and accepted deontic powers?"

If the answer to this question is positive, then W is an institution.

3.7 Institutional Robotics

Institutional robotics [Silva, 2007, Silva and Lima, 2007] aims to answer the need for a unifying concept from social sciences and the need for inclusion of coordination devices in multi-robot systems, in order to to provide a comprehensive strategy for specifying complex social interactions among a team of robots. To that end, it combines the notions of institution [Searle, 2005, Hodgson, 2006], environment [Weyns et al., 2005b], and coordination artifact [Tummolini and Castelfranchi, 2006]. We have already discussed institutions in the previous sections.

One of the propositions of institutional robotics is to take into account the active character of the environment. Some environmental processes can change their own state independently of the activity of any agents. A ball rolling down a hill is a possible example. But also multiple agents acting in parallel can have effects on the environment, unforeseeable by any one agent (a river can be poisoned by a thousand people depositing a small portion of a toxic substance in the water, even if each individual portion is itself innocuous) [Weyns et al., 2005a].

This is also true in social environments. For instance, if nine out of ten of the clients of a bank decide to draw all their money at the same date, bankruptcy could be the unintended effect. In current robotics research, social environments are poorly modeled, which may lead to problems in real world implementations. Institutional robotics aims to overcome this flaw by considering robots not only inserted in physical environments but also in institutional environments.

Coordination artifacts are artifacts shaped for coordinating agent's actions. An artifact is something done by an agent to be used by another (or the same) agent. Typically, when we think of artifacts, we consider them as having a physical presence, but artifacts can be completely dematerialized. Social conventions and norms are examples of coordination artifacts with no physical presence. The use of a coordination artifact by an agent results in a coordinated action, but is still a single-agent action.

However, we are interested in the coordination of a team of robots, so we must consider multi-agent coordinated actions. According to Proposition 6 of [Tummolini and Castelfranchi, 2006], "there exist some artifacts such that the recognition of their use by an agent and the set of cognitive opportunities and constraints (deontic mediators) are necessary and sufficient conditions to enable a multi-agent coordinated action". Institutions can be considered artifacts of this type. Opportunities and constraints create sufficient conditions for action execution. When these conditions are associated with a deontology (obligations, permissions, etc), the actions executed follow a set of generally recognized and accepted constitutive rules. This proposition takes institutional actions as multi-agent coordinated actions performed by a single-agent.

Take the example of a priest performing a marriage. A group of agents recognizes that an agent is playing a role (the actual coordination artifact) as a priest, and accepts that he has the artificial power of doing the multi-agent coordinated action of marrying a couple. The single-agent action of an agent playing a role is the vehicle action for a collective action, like flipping the switch is the vehicle action for the supra-action of turning the light on. To marry a couple the priest must perform a certain set of bodily movements counting as marrying. That set of movements is the vehicle action for the supra-action of marrying John and Mary, which is a multi-agent coordinated action.

Following these notions, we present the institutional approach as a list of points (or assumptions) that formulate the basis of a coordination scheme under the institutional economics inspiration (a more detailed list is presented in [Silva and Lima, 2007]):

1. the coordination system is a network of institutions;
2. institutions are coordination artifacts of different types (organizations, teams, hierarchies, conventions, norms, social roles, behavioral routines, stereotyped ways of sensing and interpret certain situations, material devices and particular organizations of the physical world), that may be implemented as material objects and/or mental constructs;
3. institutions are generic, they are not designed for any specific set of robots;
4. robots are able to modify, at some extent, the material organization of their physical environment (and so modify the material basis of the institutions);
5. robots are able to deliberately modify the institutional environment: "institutional imagination" ("thought experiences" about possible outcomes of modifying current institutions) and "institutional building" (collective decision-making processes to modify "constitutional rules" of current institutions) are possible mechanisms to do so;
6. there are non-deliberate means of institutional evolution: institutions can be modified by accumulating small modifications initiated by some robots and not opposed by others;
7. robots with institutional building capabilities need a high degree of autonomy, being able to pursue their own goals grounded on their "struggle for survival" (some form of homeostasis for artificial agents);
8. every robot is created with links to some subset of the institutional network.

Institutions are the central element of the approach and the answer to the need for an unifying concept from social sciences and the need for deliberately set up coordination devices. In Chapter 4 we will present our formalization of the concept for use in robots.

Summarizing, 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. With this heuristic, the institutional approach to multi-robot systems aims to provide appropriate conceptual means to the current methodological needs of designing coordination algorithms for multi-robot systems that take into account complex social interactions.

Summary

In this chapter we describe the institutional robotics approach and its inspiration in social sciences, namely in institutional economics. We initially present two other approaches to the coordination of distributed robotic systems, self-organization and market-based algorithms, and discuss why such approaches are not suitable to the goal of considering complex social interactions in such systems. We give a brief overview of what is institutional economics and how its concepts might be useful in robotics. In order to better understand the central concept of institutional economics – institutions, we briefly describe one of its more recent branches, new institutional economics, and how a more formal approach to institutions might be helpful to our endeavors. We analyze in more detail John Searle's ontological definition of institutions. In conclusion, we describe the institutional robotics approach based on concepts presented during the chapter.

Part II

Control and Modeling Methodology

Chapter 4

Institutional Agent Controllers Based on Petri Net Models of Institutions and Institutional Environments

In this chapter, we focus on formalizing the central concept of institutional robotics - institutions. Institutions can be viewed as coordination artifacts encapsulating relevant behavioral rules for robots (possibly designed based on problem-domain knowledge), that specify social interactions of different types among actors in a given scenario (robots, humans, software agents, etc.). Each institution is modular, representing one desired behavioral rule, and allowing the construction of a complex robot behavioral controller by the composition of several institutions. They represent the basic building blocks for creating cooperative working environments for social robots and humans. Modularity allows us to distinguish social interactions and to differentiate them from interactions with the environment, contributing to a richer social environment for multi-robot systems and their interactions with humans.

We formalize institutions using an abstract representation, allowing their design and execution for distributed robotic systems, so as to obtain behaviors capturing the social interactions of interest. The use of an abstract representation combined with the intended modularity enables institutions to be shared by robots and to coordinate the collective behavior. For instance, if an institution is modified during execution, either by cooperative decision-making or by designer intervention, this change can be propagated to the entire robotic team without the need of stopping execution to re-implement the controllers.

In order to accomplish this objective we propose to use Petri Nets as an abstract representation for institutions. We present an extension, designated *executable Petri nets*, that takes into account robot actions and sensor readings, in order to design institutions that can be used by robots. Finally, our method algorithmically composes a set of institutions, to create an institutional robot controller able to execute a desired task and observe the specified social interactions.

This chapter is organized as follows. In Section 4.1 we discuss related work, focusing on coordination artifacts, electronic institutions, and Petri net plan representation. Petri nets basics are introduced in Section 4.2, followed by our executable Petri nets extension in Section 4.3. In Section 4.4 we introduce our definition of institutions and, in Section 4.5, we present how to represent them in a hierarchical fashion. Finally, in Section 4.6 we present how to obtain an institutional robot controller from a set of institutions.

4.1 Related Work

4.1.1 Coordination Artifacts

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 within the institutional robotics framework, [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.

Nevertheless, this assumption is not incompatible with the idea of institutions as external artifacts, possibly as information structures attached to physical world objects. An example of such use could be a norm that, when robots navigating a road got near a roundabout, would be wirelessly transmitted and integrated into the robots' controller, providing information on how to traverse the obstacle following a social construct.

4.1.2 Electronic Institutions in Multi-Agent Systems

Electronic institutions [Esteva et al., 2001] are institutionalized electronic organizations, composed of a vast amount of heterogenous (human and software) agents playing different roles and interacting by speech acts. They provide a computational analogue of human organizations in which intelligent agents interact to accomplish their individual and collective goals. Introduced in the work of Esteva and colleagues [Esteva et al., 2000, Esteva et al., 2004], electronic institutions are specified with a formal language [Esteva et al., 2002] and used in multi-agent systems, in particular applied to auctions in electronic commerce [Esteva and Padget, 2000] or even remotely-handled fresh fish markets [Cuni et al., 2004], possibly through a dedicated middleware implementation [Sierra et al., 2004].

Focusing on the macro level (societal) aspects of agents rather than on the internal workings, electronic institutions are considered in a dialogic framework, where agents interact only through *speech acts* [Searle, 1965]. This multi-agent systems view of interactions is far removed from that of multi-robot systems, where embodiment plays a crucial role in all interactions. Nevertheless, some of the inspiring principles for electronic institutions are also relevant to our formalization of institutions. For instance, *heterogeneity* among interacting agents must be considered and supported by institutional formalisms.

The use of the institutional paradigm in multi-agent systems continues to appear in more recent research. In [Aldewereld et al., 2010], authors aim to improve the relation of the abstracted concepts of a specified multi-agent organization to the concrete ones that appear in practical situations. Their solution relies on *count-as* statements (constitutive rules used for the assignment of status function, as discussed in Section 3.6), which, by defining the social reality, provide concrete concepts with their institutional and organizational meaning. Searle's ontology of institutions is also present in [Brito et al., 2013], where authors propose a specification language for multi-agent systems where institutional facts can be distinguished from brute facts about the environment.

4.1.3 Plan Representation with Petri Nets

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., 2011]. Therein, properties of safety and liveness of Petri nets 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 for plan representation, execution, and modeling, was introduced in [Costelha and Lima, 2012]. Task plans and robot actions are represented using Petri nets. These nets can be composed in a hierarchical fashion with Petri net and Generalized Stochastic Petri Net (GSPN) models of the environment to allow for closed-loop quantitative and qualitative analysis of the system.

The hierarchical methodology presented in Costelha's and Lima's work enables organizing separately the interaction between different task plans executed by one robot and the more detailed implementation of such plans in terms of robots actions. While the former is achieved in a higher layer, robot actions and environmental processes can be described in lower layers and represented on the above layers by means of macro places. By using an expansion algorithm, all layers can be combined into a single Petri net representation that can be used either for execution or modeling of the system. In our work we will take inspiration from this hierarchical approach to design our institutional controllers.

4.2 Petri Nets

Starting from the concept of institutions as coordination artifacts, our goal is to define them using a formal representation, leading to a standard design and execution platform. Considering the three main properties of coordination artifacts mentioned in Section 4.1.1 - specialization, encapsulation, and inspectability - we propose to use Petri nets as the formal framework to define our approach to institutions.

4.2.1 Petri Nets Notation and Definition

Introduced by Carl Adam Petri in [Petri, 1966], Petri nets are a formal modeling tool for discrete event systems. Since being introduced, abundant theoretical and practical advances have been made [Murata, 1989, Ajmone Marsan et al., 1995, Bause and Kritzinger, 2002, Girault and Valk, 2003, Cassandras and Lafortune, 2008, David and Alla, 2010]. We

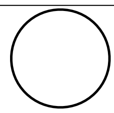
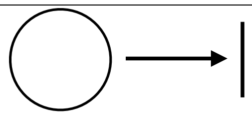
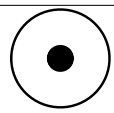
Petri Net Components	Graphical Representation
Place p	
Transition t	
Arc (p, t)	
Token in place p	

Table 4.1: Petri net components and their graphical representations

will not provide an exhaustive review of such advances, as most of them fall outside the scope of this work, apart from those relevant to the development of our methods.

We follow the definitions for Petri Nets and their dynamics present in [Cassandras and Lafortune, 2008]:

Definition 1 *A* marked 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;*
- $w : A \to \mathbb{N}^+$ *is the* weight function *on the arcs;*
- $X = [x(p_1), \ldots, x(p_n)] \in \mathbb{N}^n$*, where* $n = |P|$*, is a* marking *of the set of places* P*, representing the* state *of the Petri Net;* X_0 *is the initial marking of the net.*

Henceforth, we will refer to marked Petri nets simply as Petri nets. 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.

Table 4.1 displays the graphical representation of each of the Petri net tuple components. Fig. 4.1-(a) displays an example of a Petri net with four places and three transitions and where all arcs have weight one. 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 $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$, respectively.

State transitions in Petri Nets occur by moving tokens through the net and changing the marking by doing so. Note that this does not mean that the number of tokens in the net is conserved, rather it can be increased, maintained or decreased, depending on state changes. Petri Net state dynamics are provided by the following state transition function:

Definition 2 *The* state transition function*,* $f : \mathbb{N}^n \times T \to \mathbb{N}^n$*, of Petri Net* (P, T, A, w, X) *is defined for marking* X *and transition* t_j *if and only if*

$$x(p_i) \geq w(p_i, t_j) \quad \textit{for all } p_i \in I(t_j) \tag{4.1}$$

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

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i), \quad i = 1, \ldots, n \tag{4.2}$$

If transition t_j verifies condition (4.1) then we say it is *enabled* at marking X. 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 (4.2).

In Fig. 4.1-(b) and -(c), we display a sequence of markings of the example Petri net, obtained from the initial marking $X_0 = [1\ 0\ 0\ 0]$ displayed in Fig. 4.1-(a) through the firing of transitions. In Fig. 4.1-(a) the only enabled transition is t_1, since it is the only that has all its input places marked. The firing of transition t_1 triggers a change in the state of the Petri net, creating a new marking $X_1 = f(X_0, t_1) = [0\ 1\ 1\ 0]$, displayed in Fig. 4.1-(b). In this marking, t_1 is no longer enabled but t_2 is. Firing of transition t_2 leads again to a change in the state, creating a new marking $X_2 = f(X_1, t_2) = [0\ 1\ 0\ 1]$, displayed in Fig. 4.1-(c). In this marking there are no enabled transitions, meaning it is a *deadlock* marking of the net.

Not all states in $\mathbb{N}^n$ can be reached from the initial marking X_0 of a Petri net. This can be observed in the example of Fig. 4.1, for instance state $[0\ 0\ 0\ 1]$ can not be reached from the initial marking of the net. Thus, there is a need to define what states of a Petri net can be reached. To do so, we must first extend the state transition function f to deal with sequences of transition firings:

$$\begin{aligned} f(X, \varepsilon) &:= X & (4.3)\\ f(X, st) &:= f(f(X, s), t), \ \textit{for } s \in T^* \textit{ and } t \in T, & (4.4) \end{aligned}$$

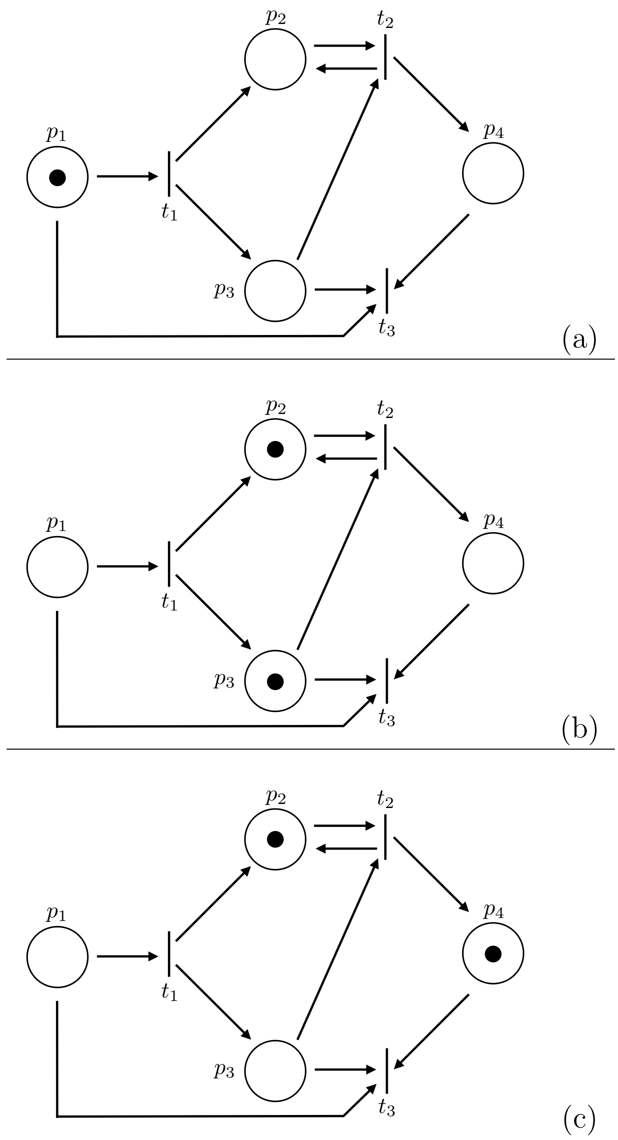


Figure 4.1: Example of the graphical representation of a Petri net. All arcs displayed have weight one. (a), (b), and (c), display a sequence of markings of the Petri net, starting with the initial marking $X_0 = [1\ 0\ 0\ 0]$, and with subsequent markings being reached by the firing of transitions t_1 and t_2.

where ε is interpreted as the absence of transition firing. We can now define the set R of *reachable states* of the Petri Net (P, T, A, w, X) as:

$$R[(P, T, A, w, X)] := \{Y \in \mathbb{N}^n \ : \ \exists s \in T^* \ (f(X, s) = Y)\} \tag{4.5}$$

The set of reachable states can be used to examine several Petri nets properties, such as the existence of possible deadlocks in the net. These properties can also be examined through a large body of analysis techniques that have been developed to study Petri nets. These properties can be of a qualitative nature, as the mentioned existence of deadlocks, or of a quantitative nature, for instance determining the maximum marking of a given set of places. We will describe two important qualitative properties of Petri nets that are required for the Petri net extension to be presented in the next section: boundedness and liveness (both definitions obtained from [Cassandras and Lafortune, 2008]).

Definition 3 *Place $p_i \in P$ in Petri net (P, T, A, w, X_0) is said to be* k-bounded, *or* k-safe, *if $x(p_i) \leq k$ for all states $X \in R[(P, T, A, w, X_0)]$.*

If a place is 1-bounded, it is called *safe*. If all places in the net are safe, the Petri net itself is called safe.

Definition 4 *Petri net (P, T, A, w, X_0) is said to be* live *if there always exists some sample path (sequence of transitions, $s \in T^*$) such that any transition can eventually fire from any state reached from X_0.*

Both these properties can be examined using *coverability tree* analysis techniques. After obtaining the coverability graph for a given net, the verification of boundedness is straightforward, while verifying liveness can be done either by brute force approach or by application of more complex analysis techniques [He and Lemmon, 2000, Esparza, 1994].

4.2.2 Choice of Petri Nets Formalism

Our choice of Petri Nets as the formal tool to define institutions 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. This property endows Petri nets with a modular and compositional nature, relevant to our efforts since we are interested in modeling institutions as distinct modules, and combine them into a robotic controller. This modular nature allows for the specialization property of coordination artifacts to be implemented.

The large body of work available concerning Petri net properties, and relevant analysis techniques to verify them, is also an advantage. Being able to quickly verify properties as boundedness and liveness is fundamental in guaranteeing our controllers perform as expected. The encapsulation property of coordination artifacts can be associated with the liveness property of Petri nets. If, as said before, encapsulation is the key to achieve reuse of coordination, guaranteeing that such artifacts are free of deadlocks ensures us that they can be reused.

It must also be noted that the Petri nets graphical representation is not only intuitive, but captures a great deal of structural information about the system. Such representation allows for simple design of coordination artifacts with Petri nets, guaranteeing also that such artifacts can be inspected in order to correct any errors present. A further relevant property is that although composition of Petri nets leads to exponential growth in the state space [same as Finite State Automata (FSA)], it allows a linear increase in the size of the graphical representation. Thus, the last property of coordination artifacts, inspectability, can also be verified.

Finally, Petri Nets also have a larger representational power than FSA, being able to mark, with finite structure, non-regular languages, i.e., languages that are not marked by FSA [Cassandras and Lafortune, 2008].

4.3 Executable Petri Nets

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 actions (combinations of two or more primitive actions can be considered as actions, depending on the degree of granularity desired by the designer). Primitive actions are those that are executed via a single actuator command.

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 an extension of Petri Nets used for design and execution of robotic controllers.

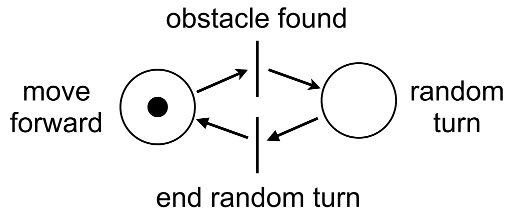


Figure 4.2: Example of an executable Petri net, specifying a simple behavior for obstacle avoidance. Places are associated with actions, transitions are associated with boolean conditions.

Definition 5 *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* $pa_i \in Pac$*;*
- $w(p_i, t_j) = 1$, $w(t_j, p_i) = 1$, *for all* i, j, *such that* $(p_i, t_j) \in A$ *or* $(t_j, p_i) \in A$*;*
- *the Petri net* (P, T, A, w, X) *is safe and live.*

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.

Since we are associating marked places with actions to be executed we want to restrict the marking of the places to one token. Multiple tokens in a given place could be interpreted as an action being executed multiple times at the same time step, which does not make sense in a robotic controller formulation. This means we must verify that the net is safe (1-bounded). We restrict EPNs to nets where all the arcs have weight 1. Although this does not guarantee the necessary boundedness, it does simplify the design process.

Verifying that the net is live ensures us that there are no deadlocks and the net is "reusable" (from a coordination artifacts perspective). From any reachable state, any transition can still fire, meaning the robot controller can always return to its initial state. This allows the EPN to be executed multiple times by a robot.

In Fig. 4.2 we display an example of an EPN, specifying a very simple obstacle avoidance behavior. Places are associated with actions the robot can execute, for instance "move forward" and "perform a random turn". Transitions are associated with boolean conditions the robot can verify using onboard sensors, irrespective of these being proprioceptive or exteroceptive. For instance, condition “obstacle found” can be verified using exteroceptive sensors such as proximity sensors, and condition “end of random turn” can be verified using propioceptive sensors such as motor encoders. This trivial net is both safe and live.

Algorithm 1 Execute Petri Net: algorithm to be executed by the robot at each time step in order to perform the behavior specified in EPN (P, T, A, w, X_0). Parameters:
EPN (P, T, A, w, X_0);
actions $a_i \in Act$, $i = 1, \ldots, n$;
conditions $c_j \in Cdt$, $j = 1, \ldots, m$, where $m = |T|$;
parameter actions $pa_j \in Pac$, $j = 1, \ldots, m$

repeat
  for all enabled transitions $t_j \in T$ **do**
    if associated condition c_j is true **then**
      run associated parameter action pa_j
      fire transition t_j
    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

Algorithm 1 is performed by the robots at each time step, allowing the robots to execute the behavior designed in an EPN by executing the actions associated with marked places. The execution of different actions at the same time step is considered to be parallelizable but verification is left to the designer of the EPN. Referring to the example of Fig. 4.2, the robot would execute the “move forward” action until the detection of an obstacle. At this point, condition “obstacle found” being verified would trigger a change in the state of the net, through the firing of the associated transition. Action “move forward” would no longer be executed since the associated place would not be marked, but action “random turn” would be executed, since its place would be marked. Upon condition “end random turn” being verified the net would revert to its original state.

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 for two properties that are of relevance to our IR approach. First, it 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. Second, it allows the robots to view the EPN as a purely abstract entity, enabling the sharing of EPNs among robots without the sharing of the actual implementation of all the actions therein represented.

4.4 Institutions

A previous abstract definition of institution was presented in [Silva et al., 2008]. There, the authors define institution as a tuple *(ID, Rationale, Modifiers, Network, Institutional Building, History)*, where each element of the tuple attempts to capture the main constitutive elements of the social order dynamics. Institutions are designated as nodes of an institutional network. For our purpose of formalizing institutions using an abstract representation, allowing for a standard design and execution platform, this definition is not sufficient, due to the general form in which it is presented. However, the notion of institutional network will be useful.

Our goal is to formalize institutions as coordination artifacts in a modular fashion. We represent each institution by an EPN that can be executed independently or together with other institutions. We also represent robot's *individual behaviors* by EPNs. While the institutions specify behaviors that have a *social nature*, i.e., they relate the robot to other robots in some way, the individual behaviors specify a set of basic behaviors that have exclusively an *individual nature*, i.e., they relate the robot with the surrounding environment and its own goals. The composition of the individual behavior with a set of institutions generates a robot controller.

The distinction between institution and individual behavior is a design choice, based on the differentiation between social interactions and interactions with the environment. Individual behaviors specify basic behaviors, closely related to the robots own goals and their "struggle for survival", and initially set by the designer. Even if individual behaviors produce some interactions with other robots, these are not planned and result from the dynamic nature of the environment. On the other hand, institutions specify socially constructed behaviors that further help robots achieve their goals and coordinate with the rest of the team. Interactions through institutions are distinguished as social interactions and the robots may deliberate how to change or if to conform to them.

It must also be taken into account that institutions are not executed continuously. For instance, when a person is driving and observing the road code, it does not simultaneously perform the necessary steps to open a bank account. Not only are these two distinct institutions, but also their concurrent execution is impossible for the person. Execution of institutions should be triggered (and also halted) by environmental conditions, present either in the physical environment or in the institutional environment. We say that an institution can be in either an *active* or *idle* state of execution.

The composition of the individual behavior with a set of institutions is non-trivial, exactly because 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 the United Kindgom, and thus should not be executed when in that territory.

In [Crawford and Ostrom, 1995], the authors 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 (a term used to describe a generalization of rules, norms, and strategies). The goal was to deal with the problem of how to specify when institutional statements can or cannot be followed. In our formalization, such operators affect whether institutions are active or idle. Loosely, they specify if an institution *can* be active (P), *must* be active (O), or *must* be idle (F).

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 the United Kingdom is active. This referencing of other institutions creates a problem for our intended modular approach to the formalization of institutions. 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 robot behaviors and other institutions in general).

Definition 6 *The set D_I of deontic operators for institutions includes the following deontic operators: $D_I = \{AllowAll,\ StopInd,\ StopInst,\ StopAll\}$. 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.*

This set of deontic operators is more restrictive exactly because it avoids making references to any specific institutions. For instance, when an institution cannot be executed concurrently with another institution, we must specify that it cannot be executed with any other institution. This restrictive specification is the price to pay for having a general and modular formalization. One possible solution to this problem is to introduce a classification of different institutional forms (organizations, norms, hierarchies, roles, etc.) and consider deontic operators that take into account these forms. However, such a solution would put extra effort in the design of institutions, moving away from the objective of minimizing designer intervention, so we chose to avoid it.

We now present our formalized definition of institution:

Definition 7 *An* Institution I *is a four-tuple* ($Inst$, $initial_I$, $final_I$, d_I) *where:*

- $Inst$ *is an EPN* (P, T, A, w, X_0) *with associated actions, conditions and parameter actions;*
- $initial_I$, $final_I \in Cdt$ *are initial and final conditions for the execution of* $Inst$*;*
- $d_I \in D_I$ *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 activation and idling are dictated by conditions $initial_I$ and $final_I$, which the robot verifies at each time step. These control when the institution is active or idle. Institutions also include a deontic operator d_I which is used when combining it with the robot individual behavior and further institutions. $Inst$ must be designed, but institutions can be kept simple and further behavioral complexity is the result of composition, in a modular fashion.

An example of a possible combination of individual behavior and institutions for a team of robots operating in a real environment is the following. The main task of the robots is to obtain a map of a given area populated with humans carrying out some activity. This can be characterized as the individual behavior of the robots. It relates the robots only to the environment and has no explicit need for coordination among robots.

A first institution implements a behavior that enforces constraints on navigation (for instance, reduced speed and no sudden changes in direction) when in the proximity of humans. This behavior is of a social nature, relating robots to the human actors working in the environment. Initial and final conditions are the detection of a human entering or leaving a radius of a given distance around the robot. The deontic operator associated with such institution is $AllowAll$, specifying that this behavior can be executed concurrently with others, for instance the mapping of the environment.

A second institution implements an emergency behavior, where robots must coordinate among themselves to quickly navigate in an established formation (e.g., line formation)

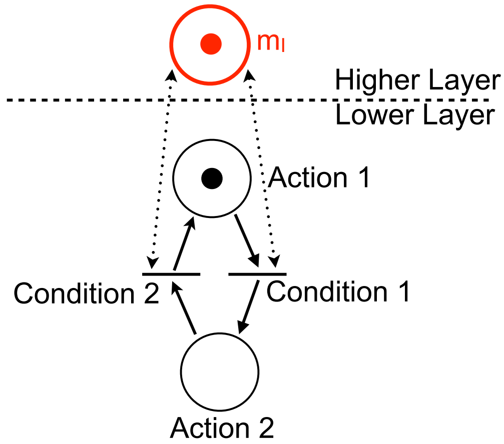


Figure 4.3: Hierarchical representation of an EPN in two layers. Dotted arcs represent two directional arcs, one from a transition to a place and one from a place to a transition. Lower layer, EPN with conditions and actions associated to transitions and places. Higher layer, macro place m in red.

to a given area of the environment where they do not impact the safety of human actors. This behavior also has a social nature, since robots must coordinate among themselves (and possibly with humans). Initial and final conditions are the detection of an emergency signal (for instance, a fire or an alarm ringing) and the reception of an order from a human actor to abandon the emergency behavior, respectively. The deontic operator associated with this institution is *StopAll* since in such a critical situation it must be guaranteed that no other behaviors are executed concurrently.

The necessary regulation among behaviors is obtained through the composition of their associated EPNs, guided by the deontic operators of institutions, resulting in a controller to be executed by each individual robot. The composition process will be tackled in the subsequent sections.

4.5 Hierarchical Representation of Executable Petri Nets

As previously mentioned, Petri Nets (and thus EPN) can be represented by macro places in a hierarchical fashion, using two distinct layers. In our approach we consider that individual behaviors and institutions are part of a lower layer and are represented by one macro place in the higher layer, as shown in Fig. 4.3. In the lower layer, an abstract EPN is displayed. This EPN represents either an individual behavior *Ind* or an

EPN $Inst$ of an institution I. Note that the example EPN structure displayed does not represent any template that must be followed for the design of individual behaviors and institutions.

In the higher layer of Fig. 4.3, the macro place m representing that same EPN is displayed. By adding arcs from each transition in the EPN to m and from m to each transition (shown as a single bidirectional dotted arc), we guarantee that each transition will only be enabled if m is marked. When a transition in the EPN fires, m will continue to be marked since it is a output place of the transition.

Thus, if a higher layer macro place is marked, the lower layer EPN of 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 to obtain a full EPN that can be used as controller.

4.6 Institutional Agent Controller

In this section we will introduce the concept of institutional agent controller, the robotic controller obtained from the composition of a set of institutions and individual behavior, that will be executed in robots so as to obtain behaviors capturing the social interactions of interest.

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. 4.4-(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_{Ii} is marked. Since only one place from the set m_{Ii} and $idle_{Ii}$ can be marked at each time, we have that institution I_i is active if m_{Ii} is marked and idle if $idle_{Ii}$ is marked. This allows us to control the activation and idling of institutions with their initial and final conditions as shown in the Fig. 4.4-(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. 4.4. 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 $AllowAll$.

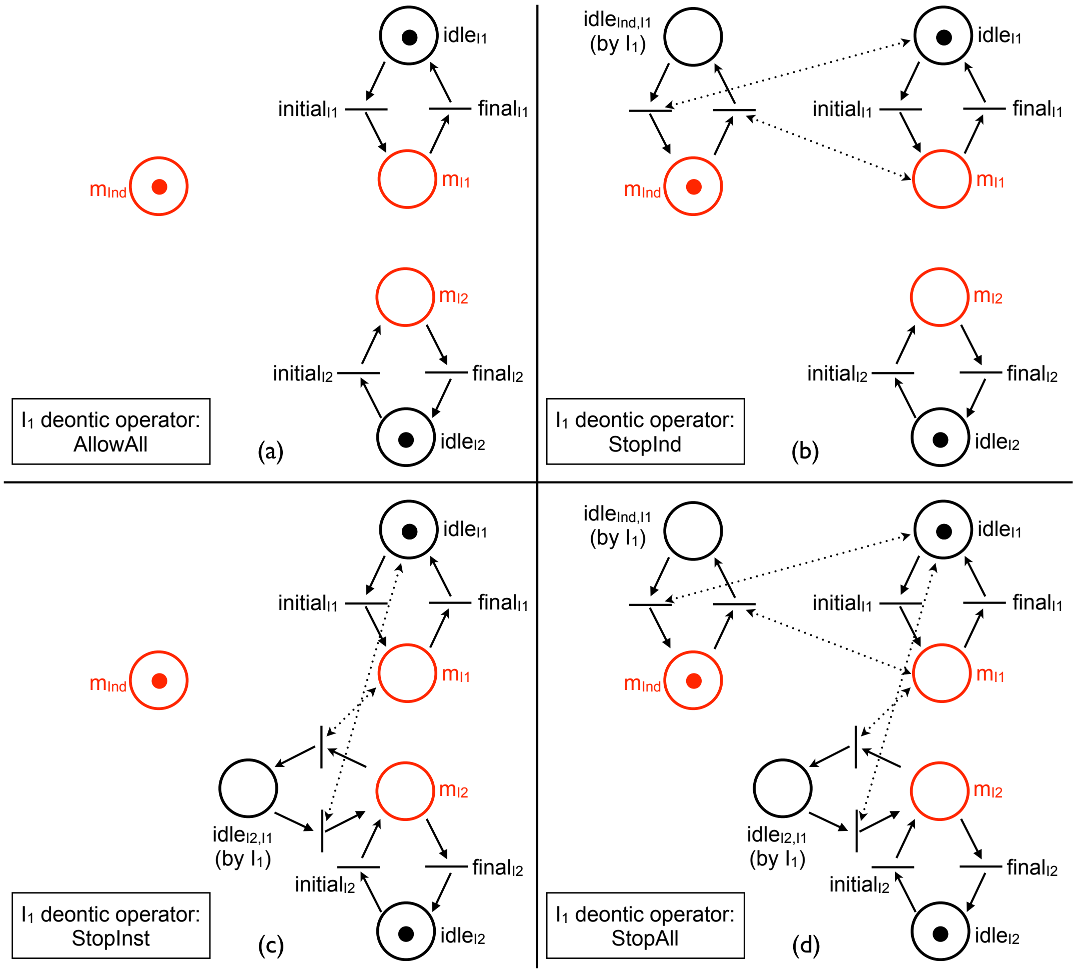


Figure 4.4: Higher layer composition scheme for two institutions I_1, I_2 and individual behavior Ind. Dotted arcs represent bidirectional arcs, as in Fig. 4.3. Places in red are macro places representing implementations of institutions and the individual behavior in the lower layer. (a) composition 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 $StopAll$.

A detailed algorithm for composing institutions with different deontic operators will be presented subsequently.

If the deontic operator of institution I_1 is also *AllowAll* (Fig. 4.4-(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. 4.4-(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, which causes them to fire immediately when their inputs places are marked, thus eliminating any conflicts with other simultaneously enabled transitions. This specifies that if institution I_1 is activated, then the individual behavior Ind is set to idle and, when I_1 is idled, Ind is set back to active.

If the deontic operator of I_1 is *StopInst*, as in Fig. 4.4-(c), the same structure is added, but now with respect to the macro place 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 execution 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. 4.4-(d). These rules would be applied in the same manner if institution I_2 had a different deontic operator. In such case, the rules would also be applied for institution I_2, adding structures to the net depending on its deontic operator.

We can now define an Institutional Agent Controller that can be executed by robots:

Definition 8 *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 $d_{I_1}, \ldots, d_{I_n}$.*

The IAC can be obtained through Algorithm 2. Fig. 4.5 displays the graphical representation of an abstract IAC resulting from the composition of one institution I (with deontic operator *StopInd*) and an individual behavior Ind. All macro places and control places ($idle_x$) added during composition are associated with a void action. Control transitions added during composition are associated with a special condition that is always true (except when associated with initial or final conditions of institutions). In order to guarantee that the IAC resulting from the composition procedure is indeed an EPN we must prove that such procedure preserves the safeness and liveness properties associated with EPNs.

Proposition 1 *The composition procedure described in Algorithm 2 preserves the safe-*

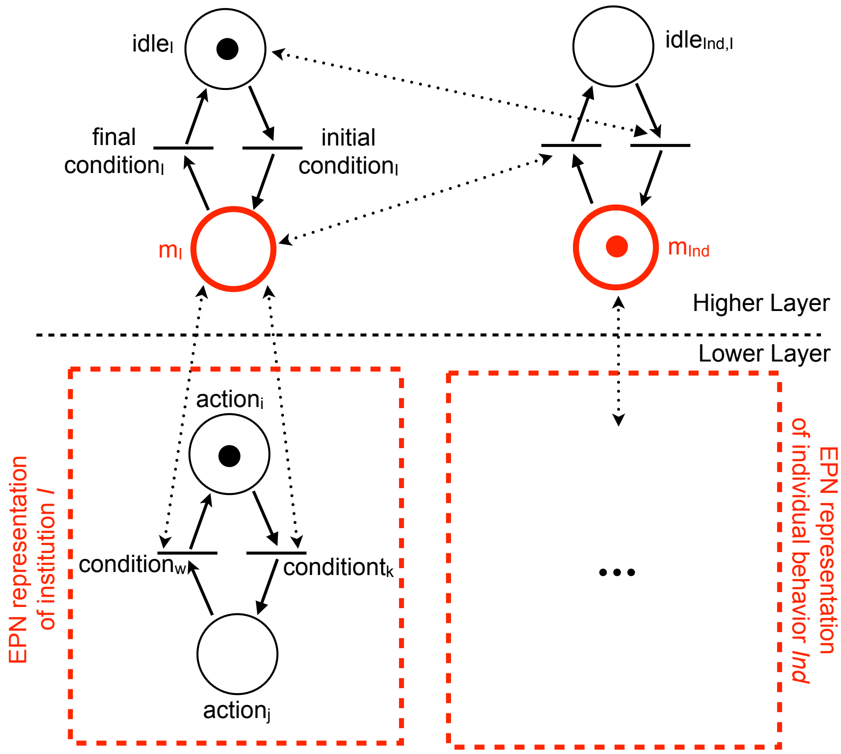


Figure 4.5: Abstracted Institutional Agent Controller graphical representation. Lower layer: modular EPN representation of institution I and individual behavior *Ind*. Higher layer: regulatory places and transitions added during IAC composition.

ness and liveness properties of Ind *and* $Inst_1, \ldots, Inst_n$ *(*$Inst_i$ *is the EPN part of institution* I_i*), resulting in an IAC which verifies the same properties.*

The proof for this proposition is presented in Appendix A.

Considering the associations between places and actions, and transitions and conditions, and Proposition 1, 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. Line 9 should be replaced by "for all marked places $p_i \in P$ for which m_j is marked do", where m_j is the corresponding higher layer macro place. The effort needed for the formalization includes the design effort of the institutions and individual behavior and composition effort. 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 (roughly the same as with an FSA).

Algorithm 2 Composition: taking as input an individual behavior and a set of institutions, this algorithm returns an IAC that can be executed by robots
Parameters:
individual behavior Ind;
institutions $\{I_1, \ldots, I_{N_I}\}$, where N_I is the number of considered institutions and $I_i = (Inst_i, initial_{I,i}, final_{I,i}, d_{I,i})$.

create an empty EPN, denoted IAC_{net}
{We start by adding the lower layer EPNs to the output net.}
add individual behavior net, Ind, to IAC_{net}
for $i = 1 \to N_I$ **do**
  add institution i net, $Inst_i$, to IAC_{net}
end for
{The next step is to add the higher layer representations of the EPNs we just added. For institutions we also add idle places, and initial and final conditions that guide activation and idling of institutions.}
create macro place m_{Ind}, place token in m_{Ind}
for all $t \in T_{Ind}$ **do**
  {T_{Ind} is the set of transitions of Ind.}
  add bidirectional arcs between transition t and macro place m_{Ind}
end for
for $i = 1 \to N_I$ **do**
  create macro place m_{Ii}
  for all $t \in T_{Inst,i}$ **do**
    {$T_{Inst,i}$ is the set of transitions of $Inst_i$.}
    add bidirectional arcs between transition t and macro place m_{Ii}
  end for
  create control place $idle_{Ii}$, place token in $idle_{Ii}$
  create transition with input place $idle_{Ii}$ and output place m_{Ii}, and associated with condition $initial_{I,i} \in Cdt$
  create transition with input place m_{Ii} and output place $idle_{Ii}$, and associated with condition $final_{I,i} \in Cdt$
end for
{The final step of the algorithm uses the deontic operators of each institution to create control places and transitions that prevent the undesired concurrent execution of certain behaviors.}
for $i = 1 \to N_I$ **do**
  if $d_{I,i} = AllowAll$ **then**
    continue to next i
  end if

if $d_{I,i} = StopInd \ \vee \ d_{I,i} = StopAll$ **then**
create control place $idle_{Ind,Ii}$
create control transition with input places m_{Ii} and m_{Ind}, and output places m_{Ii} and $idle_{Ind,Ii}$
create control transition with input places $idle_{Ii}$ and $idle_{Ind,Ii}$, and output places $idle_{Ii}$ and m_{Ind}
end if
if $d_{I,i} = StopInst \ \vee \ d_{I,i} = StopAll$ **then**
for $j = 1 \rightarrow N_I, \ j \neq i$ **do**
create control place $idle_{Ij,Ii}$
create control transition with input places m_{Ii} and m_{Ij}, and output places m_{Ii} and $idle_{Ij,Ii}$
create control transition with input places $idle_{Ii}$ and $idle_{Ij,Ii}$, and output places $idle_{Ii}$ and m_{Ij}
end for
end if
end for
return IAC_{net}

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, or even new institutions being added during execution. Thus, the method is fully scalable to any number of robots. Complexity of the IAC increases only with the number of institutions.

Summary

In this chapter 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 herein. We defined institutions and 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 algorithm 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.

Chapter 5

Probabilistic Modeling with Institutional Agent Controllers and Generalized Stochastic Petri Nets

Real world robotics requires formal methods that can accurately predict the performance of the system. While this is true for all fields of robotics, it is of noted relevance for the field of distributed robotic systems. The behavior of distributed robotic systems with a large number of robots is difficult to model, since these are, most of the times, stochastic, dynamic, and non-linear in nature. Traditionally, when implementing these systems in reality, researchers tend to use small dimensions and low cost robots, allowing a large number of robots on limited physical space and limited budget. Such robots are prone to noise in sensing and actuation, presenting another difficulty when developing models for such systems. Modeling techniques for large distributed robotic systems, capable of predicting their performance and allowing verification of relevant properties, are of critical importance. They allow researchers to test a broad range of parameters and design choices that would take too long to test with the large number of robots considered.

One of the goals of our research is to develop institutional robotics models that predict the system's performance both quantitatively and qualitatively, and to analyze their intrinsic limitations, performance bounds, and general system properties (e.g., liveness). Some qualitative properties can be obtained from the EPN structure of IAC discussed in the previous chapter. In this chapter we focus on quantitative analysis of the system. We show that a Generalized Stochastic Petri Net (GSPN) view of IACs in closed loop with the distributed robotic system they control, enables quantitative model-based analysis. To do so, we will follow a multi-level modeling approach that considers models in different levels of abstraction.

Given the focus of our approach, we are interested in modeling aspects of social interaction amongst robots. In our control methodology, these are represented in the robot controller, the IAC, as institutions. As discussed, the IAC is divided in two layers, where the lower layer contains the implementations of institutions, and the higher layer describes how those institutions relate to one another. Moreover, since the IAC will be used as a starting point for our models, this hierarchical division of control will lead us to consider a hierarchical approach to modeling. Different layers will model relations between behaviors, or the actual implementation of such behaviors, or even information about the environment of the system.

This chapter is organized as follows. In Section 5.1 we introduce the multi-level modeling methodology we will follow. Sections 5.2 and 5.3 introduce GSPNs and describe how to analyze them, respectively. Finally, in Section 5.4 we introduce our hierarchical modeling approach using the IAC as a starting point.

5.1 Multi-Level Probabilistic Modeling Methodology

In our research we are interested in borrowing some concepts from a multi-level probabilistic modeling methodology established for swarm robotic systems. This methodology takes into account the individual robot controller in order to generate an aggregated macroscopic representation of the dynamics of the whole team. It was proposed by Martinoli et al. [Martinoli et al., 2004] for a collaborative swarm robotics case study and has been applied to other case studies concerned with robot aggregation [Correll and Martinoli, 2011, Evans et al., 2010] and wireless connectivity [Winfield et al., 2008]. This methodology has also been compared with other approaches based on multiple levels of modeling but proceeding in a top-down fashion in terms of model-building and control design [Berman et al., 2009, Berman et al., 2011]. As shown in [Mermoud et al., 2014], a bottom-up modeling and control design appears to be particularly indicated to deal with resource-constrained robots.

The Multi-Level Modeling Methodology (MLMM) is characterized by three major points. First, starting with a real implementation of a particular case study, the methodology builds a series of models increasing in their abstraction level. These levels are briefly described in Table 5.1. Second, a representation of the robot controller is used across all levels as a blueprint for the models. Third, a consistent set of metrics and parameters is shared by models in different levels of abstraction.

The increase in abstraction on each new level of the MLMM corresponds to a decrease in the computational cost of the model, making each new level more easy to analyze, although with a less accurate representation of the original system. The levels described

Models	Abstraction Features
Submicroscopic	Individual intra-robot component representation Parametrized/parametric noise models Kinematic/dynamic motion assumption 3D embodiement
Microscopic (spatial)	Simple kinematics Abstraction of robots' embodiment Point or simple shape representation Pose as state variable
Microscopic (non-spatial)	Representation of individual robots by probabilistic controller Geometric approximation of probabilities
Macroscopic	Aggregation of individual representations Typically mean field approach (ODE representation)

Table 5.1: Standard hierarchy of modeling levels in MLMM and assumptions taken while increasing abstraction level

in Table 5.1 can be further described as discussed below.

- *Submicroscopic*: the lower level model is typically implemented using embodied, realistic simulation tools capable of representing intra-robot details in a very accurate way (e.g., individual sensors and actuators are modeled separately with their noise characteristics, nonlinear response, placement and orientation). An example is the *Webots* simulator, described in Chapter 2. A controller designed to accomplish the underlying task of the case study, and defined in a formal way (for instance, a FSA), is executed in each individual robot. The aim of submicroscopic models is to abstract as little as possible the aspects of the real system. However, there is always a level on which reality cannot be fully represent. For instance, any group of homogenous robots is, in reality, heterogenous, since the noise profile in each robots' sensors is not exactly the same and manufacturing differences among robots cannot be replicated.

- *Microscopic (spatial/non-spatial)*: we divide this level into spatial and non-spatial since the loss of spatiality is a major abstraction for a model. Nevertheless, the underlying principle is the same, each robot is still considered as an individual entity but most of the intra-robot details are abstracted. In spatial models, the robots' embodiment is abstract, usually considering each robot as a point or simple shape on a 2D space. These abstracted robots act upon the environment following the same controller used in the submicroscopic simulations but move according to simple kinematic laws. An example of such a microscopic simulator is the *twodeepuck* simulator, also described in Chapter 2. In non-spatial models, each robot is abstracted by a probabilistic version of the controller (for instance, a Probabilistic Finite State Automata (PFSA)). At each moment in time, a robot is in one of the states of his controller, and transitions to other states are considered according to a certain probability distribution. This distribution is obtained from parameters common to other layers, the geometry of the abstracted space, or even from data gathered in the lower levels of abstraction.

- *Macroscopic*: the higher level of abstraction uses a formal representation of the controller (for instance, PFSA) to generate, in a mean field approach, an estimate of the number of robots in each state of the controller. This corresponds to aggregating the representation of all robots, which before were modeled as separate entities, into a single representation. The whole team is modeled as a single entity and its dynamics are modeled probabilistically, being represented by a Markovian or semi-Markovian process. This characterization allows the analysis of transient and steady-state properties of the system that would not be possible at lower levels of abstraction. In this work we focus on analyzing steady-state properties of the systems being modeled.

In this work we employ all the levels described above to implement and analyze our case studies. However, and since the study of the MLMM is not the main focus of our work, there is no particular case study in which all the levels are used. In Chapter 6, we validate our approach to probabilistic modeling and consider submicroscopic and macroscopic levels of the MLMM – real robot experiments, submicroscopic simulations, and macroscopic analysis. Chapter 7 only considers submicroscopic models, while Chapter 8 considers both microscopic (spatial) and macroscopic models.

In the remainder of this chapter we will describe how to apply the MLMM to our institutional robotics approach, making use of the EPN structure of the IAC, not only to control the robots at different levels of abstraction, but in particular to obtain macroscopic models of the system.

5.2 Generalized Stochastic Petri Nets

In this section we introduce the concept of Generalized Stochastic Petri Nets (GSPN) [Ajmone Marsan et al., 1984, Ajmone Marsan et al., 1995, Murata, 1989, Bause and Kritzinger, 2002], an extension of Petri nets that considers the notion of time. In Petri nets (as defined in Section 4.2), time is not taken into account, only the sequence of firing of transitions determines the state of the net. Moreover, there is no explicit way to resolve conflicts when two or more transitions are enabled, their firing is treated in a non-deterministic way. To model the quantitative performance of a given system, its

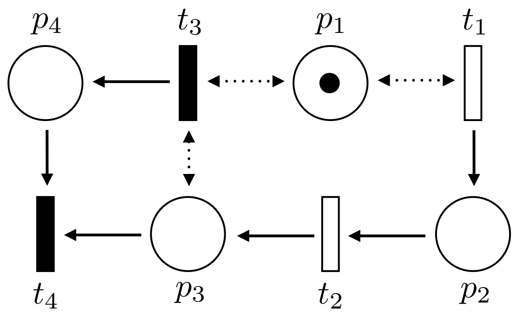


Figure 5.1: Example of a Generalized Stochastic Petri Net. In GSPNs, immediate and timed transitions are represented as full and empty rectangles, respectively. We represent bidirectional arcs as dotted, double-arrowed arcs. Initial marking $X_0 = [1\ 0\ 0\ 0]$.

temporal behavior has to be represented in the net.

In GSPNs time is taken into account in the firing of transitions. While before firing was immediate, now we consider two types of transition: *immediate* and *timed*. Immediate transitions fire in zero time, but for timed transitions a probabilistic distribution (specified for each transition, possibly marking-dependent) defines a time delay between the enabling of the transition and its firing. In GSPNs this time delay is restricted to be exponentially distributed, i.e., drawn from an exponential distribution defined by a given rate λ.

We follow the definition and notation for GSPNs and their dynamics present in [Bause and Kritzinger, 2002] (we replace w by ω to avoid notation conflicts):

Definition 9 *A* Generalized Stochastic Petri Net *is a four-tuple* (PN, T_1, T_2, W) *where:*

- $PN = (P, T, A, w, X)$ *is a marked Petri net;*
- $T_1 \subseteq T$ *is the set of timed transitions,* $T_1 \neq \emptyset$*;*
- $T_2 \subseteq T$ *is the set of immediate transitions,* $T_1 \cap T_2 = \emptyset$*,* $T = T_1 \cup T_2$*;*
- $W = (\omega_1, \ldots, \omega_m)$*,* $m = |T|$*, is an array whose entry* $\omega_i \in \mathbb{R}^+$ *is*
  - *a (possibly marking-dependent) rate of an exponential distribution specifying the firing delay, when* $t_i \in T_1$ *(in which case we rename it* $\lambda_i = \omega_i$*);*
  - or *a (possibly marking-dependent) firing weight, when* $t_i \in T_2$*.*

In Fig. 5.1, we display an example of the graphical representation of a GSPN. We follow the usual notation of representing immediate and timed transitions by full and empty rectangles, respectively. We introduce a representation for bidirectional arcs as dotted, double-arrowed arcs.

The state dynamics described in (4.1) and (4.2) in Section 4.2 are maintained. However, the sequence of firing transitions is now determined probabilistically. We denote the set of enabled transitions at marking X by $EN_T(X)$ and the probability of transition t firing at marking X by $P(t|X)$.

If only one transition t_i is enabled at marking X, then that transition fires with probability $P(t_i|X) = 1$. Depending on whether t_i is immediate or timed, it fires in zero time or with a time delay exponentially distributed with rate λ_i (the average firing time is $\frac{1}{\lambda_i}$), respectively. In the example of Fig. 5.1, only t_1 is enabled at marking X_0 so it fires with an *expected* time delay of $\frac{1}{\lambda_1}$. Note that transition t_3 is not enabled since p_3 is one of its input places and it is not marked.

If more than one transition is enabled in marking X and all enabled transitions are timed ($EN_T(X) \subseteq T_1$), the probability of transition $t_i \in EN_T(X)$ firing can be obtained from the rates of the enabled transitions in the following manner:

$$P(t_i|X) = \frac{\lambda_i}{\sum_{j:t_j \in EN_T(X)} \lambda_j} \tag{5.1}$$

This equation is derived from the exponential distributions of the time delays of timed transitions [Bause and Kritzinger, 2002]. In the example of Fig. 5.1, after t_1 fires the net marking is $X_1 = [1\ 1\ 0\ 0]$, and both t_1 and t_2 are enabled. The probability of t_2 firing before t_1 is calculated as $P(t_2|X_1) = \frac{\lambda_2}{\lambda_1+\lambda_2}$.

Assuming t_2 fires before t_1, the net moves to marking $X_2 = [1\ 0\ 1\ 0]$. At this marking, t_1 and t_3 are enabled. However, $t_3 \in T_2$ is an immediate transition. In GSPNs, immediate transitions always fire before timed transitions, by definition.

Thus, in our example, t_3 fires, moving the net to marking $X_3 = [1\ 0\ 1\ 1]$. Again, both immediate and timed transitions are enabled. As defined, immediate transitions are first to fire. When $EN_T(X)$ contains more than one immediate transition, a probability distribution for the firing of those transitions must be specified. Such distribution is called a *random switch* and is obtained from the firing weights of the enabled immediate transitions. The probability of immediate transition $t_i \in EN_T(X)$ firing is calculated in the following manner:

$$P(t_i|X) = \frac{\omega_i}{\sum_{j:t_j \in EN_T(X) \cap T_2} \omega_j} \tag{5.2}$$

The probability of t_4 firing before t_3 is calculated as $P(t_4|X_3) = \frac{\omega_4}{\omega_3+\omega_4}$.

GSPNs can be classified according to several axis dealing with the firing policy of transitions, for instance, resampling, age memory, enabling memory [Ajmone Marsan et al., 1989, Ajmone Marsan et al., 1995]. The modeling methodology we discuss in this chapter does not need to be restricted to any particular instance of GSPN formalization. The only requirement is that the net can be transformed into a Markov chain, as we will see in the next section. In Section 8.11 we will use a non-standard firing policy.

5.3 Analysis of Generalized Stochastic Petri Nets

5.3.1 Continuous Time Markov Chain Equivalence

Let us consider Stochastic Petri Nets (SPNs), a particular case of GSPNs that include only timed transitions ($T_1 = T$). Given the assumption of exponentially distributed firing delays for transitions, the probability of switching from a particular marking to another in a SPN is independent of the time spent in that marking, as we have seen with (5.1). It can also be shown that the time spent in a marking, the *sojourn* time of the marking, is exponentially distributed, with the rate of such distribution being obtained from the transition rates of enabled transitions [Bause and Kritzinger, 2002]. The combination of these two properties implies that the marking sequence of a SPN describes a Markov process.

Markov processes can be characterized in the form of Discrete Time Markov Chains (DTMC) or Continuous Time Markov Chains (CTMC) [Bause and Kritzinger, 2002], depending on how time is considered:

Definition 10 *A* Discrete Time Markov Chain *is a discrete time stochastic process* $\{\chi_n : n \geq 0\}$*, with a countable discrete state space* $\mathcal{S}$*, that satisfies the Markov property,*

$$P[\chi_{n+1} = x_{n+1} | \chi_n = x_n, \ldots, \chi_0 = x_0] = P[\chi_{n+1} = x_{n+1} | \chi_n = x_n] \tag{5.3}$$

for $x_0, \ldots, x_n, x_{n+1} \in \mathcal{S}$ *and* $n \in \mathbb{N}$.

Definition 11 *A* Continuous Time Markov Chain *is a continuous time stochastic process* $\{\chi(t) : t \geq 0\}$*, with a countable discrete state space* $\mathcal{S}$*, that satisfies the Markov property,*

$$P[\chi(t) = x | \chi(t_n) = x_n, \ldots, \chi(t_0) = x_0] = P[\chi(t) = x | \chi(t_n) = x_n] \tag{5.4}$$

for any sequence $t_0, \ldots, t_n, t$ *such that* $t_0 < \ldots < t_n \leq t$ *and* $x_0, \ldots, x_n, x \in \mathcal{S}$.

We will focus on the continuous case, since the marking process of SPNs happens in continuous time. For CTMCs, the probability of transitioning from state x_i at time t_i to state x_j at time $t_i + t_j$ is given as

$$p_{ij}(t_i, t_i + t_j) = P[\chi(t_i + t_j) = x_j | \chi(t_i) = x_i] \tag{5.5}$$

In this work we deal only with *homogenous* (or stationary) CTMCs, in which $p_{ij}(t_i, t_i + t_j)$ does not depend on t_i or $t_i + t_j$, but only the time interval t_j.

In order to analyze the CTMC, we need to obtain the transition rate matrix $Q(t)$, where each element $q_{ij}(t)$ is the rate of going from state x_i to state x_j at time t for $i \neq j$, and $q_{ii}(t)$ is the rate for leaving state x_i at time t. For homogenous CTMCs the transition rate matrix is constant over time, i.e., $Q = Q(t)$.

The analysis of SPNs is achieved by exploring their correspondence with Markov processes, by creating an associated CTMC and analyzing it. The state space $\mathcal{S}$ of the associated CTMC is the reachability set $R(PN)$ (defined in 4.5), with each marking $X \in R(PN)$ representing a state of the CTMC. The transition rate from state X_i to state X_j is given by $q_{ij} = \lambda_k$, the firing rate of transition t_k from X_i to X_j, or by the sum of rates for all transitions between the two states. $q_{ij} = 0$ if no transitions lead from state X_i to state X_j and q_{ii} is determined so as to satisfy $\sum_j q_{ij} = 0$.

However, a problem appears when considering GSPNs. The marking sequence of a GSPN does not directly describe a CTMC, since immediate transitions fire in zero time. This causes the sojourn times of markings not to be exponentially distributed. Nevertheless, the probability of changing from one marking to another is still independent of the time spent in the marking. Thus, a GSPN describes a semi-Markov process.

Consider that the firing of transitions of the GSPN happens at times $t_0, \ldots, t_n, \ldots$. If a timed transition fires at time t_i, then $t_i - t_{i-1} > 0$, since timed transitions create tangible states in which the process remains for some time (corresponding to the firing delay of the transition). However, if a immediate transition fires, then $t_i = t_{i-1}$. This means that we can think of the stochastic process associated with the marking process of GSPNs as a DTMC where some points coincide in time. This DTMC is called the Embedded Markov Chain (EMC).

Definition 12 *Given a CTMC $\{\chi(t) : t \geq 0\}$, the discrete time process $\{\chi_n : n \geq 0\}$, where χ_n denotes the state reached by the CTMC after n state transitions, is a DTMC called the* Embedded Markov Chain *of the CTMC.*

Thus, we can create a CTMC associated with the marking process of a GSPN as described above for SPNs. The calculation of the rate transition matrix now takes into account both firing rates of exponentially timed transitions and firing weights of immediate transition. The EMC associated with this CTMC has the same state space, $\mathcal{S} = R(PN)$, and its transition probability matrix P, specifying the 1-step transition probabilities from state x_i to state x_j, is given by:

$$\begin{aligned} p_{ij} &= 0, && \text{if } i = j \\ p_{ij} &= \tfrac{q_{ij}}{\sum_{k \neq i} q_{ik}}, && \text{if } i \neq j \end{aligned} \tag{5.6}$$

By analyzing the EMC we will be able to derive meaningful analysis for the CTMC and, as is our goal, for a GSPN.

5.3.2 Steady State Analysis

In this work, when stochastically modeling distributed robotic systems, we are interested solely in obtaining steady state predictions of the system. Starting with a GSPN that models the system of interest, we obtain a CTMC and the corresponding EMC as described above. We denote by $\Pi^d = [\pi_0^d(n)\ \pi_1^d(n)\ \ldots]$ the probability of reaching all the states after n time steps of the EMC have elapsed. The steady state probability distribution of the EMC, denoted by $\mathbf{\Pi}^d$, is such that

$$\mathbf{\Pi}^d = [\boldsymbol{\pi}_0^d\ \boldsymbol{\pi}_1^d\ \ldots] = \lim_{n \to \infty} \Pi^d(n), \tag{5.7}$$

and can be obtained by solving

$$\begin{aligned} \mathbf{\Pi}^d P &= \mathbf{\Pi}^d \\ \sum_{i \in \mathcal{S}} \boldsymbol{\pi}_i &= 1 \\ \boldsymbol{\pi}_i &\geq 0 \end{aligned} \tag{5.8}$$

From $\mathbf{\Pi}^d$ we can obtain the steady state probability distribution of the associated CTMC, denoted by $\mathbf{\Pi}^c$. To do so, we must take into account the time spent by the system in each state, which was not considered in the EMC. $\mathbf{\Pi}^c = [\boldsymbol{\pi}_0^c\ \boldsymbol{\pi}_1^c\ \ldots]$ is obtained from $\mathbf{\Pi}^d$ following

$$\boldsymbol{\pi}_i^c = \frac{\boldsymbol{\pi}_i^d m_i}{\sum_{j \in \mathcal{S}} \boldsymbol{\pi}_j^d m j} \tag{5.9}$$

where m_i is the average sojourn time of the CTMC state X_i. In GSPNs, states in which immediate transitions are enabled have zero sojourn time. These states are called *vanishing* and their set is denoted by $\hat{V}$. For all other states, the sojourn time is exponentially distributed, since it corresponds to the firing delay of a timed transition. These states are called *tangible* and their set is denoted by $\hat{T}$. Sojourn times for state X_i can be obtained from the transition rate matrix Q of the CTMC as follows:

$$m_i = \begin{cases} 0 & \text{, if } X_i \in \hat{V} \\ \frac{1}{\sum_{j \neq i} q_{ij}} & \text{, if } X_i \in \hat{T} \end{cases} \tag{5.10}$$

By combining (5.9) and (5.10), we obtain a steady state probability distribution for a CTMC that describes the stochastic marking process of the GSPN we intend to analyze. Using this distribution we can study several performance metrics, useful to predict the behavior of the system, such as the probability of the system being in a subset of markings, the mean number of tokens in a place, the probability of a transition firing, or the throughput at a transition. We describe the first and last of these metrics in more detail, since they will be used when modeling our case studies.

Definition 13 *Let $C \subseteq R(PN)$ constitute a set of markings of interest of a GSPN. The* probability *of the system being in any of those markings is given by:*

$$P[B] = \sum_{X_i \in B} \boldsymbol{\pi}_i^c \tag{5.11}$$

Definition 14 *The* throughput *at a timed transition t_i is given by its mean number of firings at steady state and can be calculated as:*

$$\overline{d_i} = \sum_{X_j \in EN_i} \boldsymbol{\pi}_j^c \lambda_i \tag{5.12}$$

where EN_i is the set of markings in which transition t_i is enabled.

In this work the analysis described above was implemented in the Petri Net tools described in Chapter 2. In TimeNET, such methods are also extended to deal with timed Petri nets in which both deterministically timed transitions and exponentially timed transitions are considered [German and Lindemann, 1994].

5.3.3 Generalized Stochastic Petri Net Simulation

The problem of state explosion is well known when dealing with Petri nets. Either by composition of several Petri nets, or by the simple increase of the graphical representation of the net, the set of reachable states $R(PN)$ tends to grow exponentially. Unfortunately, the EMC-based analysis of the steady state probability distribution of GSPNs becomes unfeasible with the increase in size of $R(PN)$, given a limited time window.

Nevertheless, analysis of GSPNs with a large number of states is not impossible. For instance, TimeNET provides a GSPN simulation module where such analysis can be accomplished [Kelling, 1995]. This module simulates the firing sequence of the net, randomly firing transitions according to the state of the net and the firing rates and weights of timed and immediate transitions, respectively. It allows the definition of performance measures, for instance the probability of being in a certain state or the probability that a given place has a certain number of tokens, which are recorded during the simulation. The module detects an initial transient period of the GSPN and, after that period, performs a statistical analysis on the specified measures until a certain degree of confidence provided by the user is reached, at which point the simulation is stopped.

5.4 Hierarchical Modeling with Institutional Agent Controllers

In the multi-level modeling methodology, the controller of the robots is used as a starting point for models at different levels of abstraction, including the macroscopic level. We follow this methodology, using the IAC as a starting point for a macroscopic, probabilistic GSPN model. The multi-level modeling methodology was proposed for swarm robotics systems, guided by self-organization principles. For such systems, the team performance often depends on the internal state of the robots, and especially, what fraction of the team is in those states. However, not all systems can be modeled solely using information present in the controller of the robots. In some cases, information about the environment must be modeled in order to obtain accurate predictions.

5.4.1 Multi-Layer Methodology

We propose a multi-layer hierarchical modeling methodology that uses an IAC as a starting point and adds an environmental information layer. We take inspiration from [Costelha and Lima, 2012], where a similar architecture for modeling single- or multi-robot plans, also using PNs and GSPNs, is presented. Therein, the model architecture is divided in three layers (from top to bottom): action coordinator layer, action executor layer, and environment layer. Each layer contains modular PN or GSPN blocks that are composed to produce a single GSPN. All layers are used for modeling, while the top two layers (action layers) are also used for execution of the plans.

Following the multi-level modeling methodology, we use the IAC as the starting point for our model. The IAC is divided in two layers, similar in function to the top two layers of Costelha's architecture. In both approaches the higher layer is dedicated to the specification of coordination, while the lower layer is dedicated to the specification of execution. However, the objects of such specifications are different in both approaches. In [Costelha and Lima, 2012], the objects of specification are actions, their execution modeled by GSPNs in the action executor layer and their coordination modeled in the action coordination layer. In the IAC approach, the objects of specification are behaviors, with implementation and coordination being modeled in the lower and higher layers, respectively.

We add an environmental information layer to our IAC-based GSPN model. In Fig. 5.2, we display the GSPN representation of an abstract IAC composed of one institution I and an individual behavior Ind, together with the representation of the environmental information layer. This layer contains information about certain aspects of the environment that are not present in the IAC. How to represent environmental information in the model is left open to the designer, but the GSPN structure of the model should be preserved.

In Fig. 5.2, we display two possible ways of representing such information, again taking inspiration from [Costelha and Lima, 2012]. Boolean predicate places can represent if a certain boolean predicate about the environment is true or false (true iff the place is marked). By adding a second place that is marked if the same predicate is false, we can model a transition checking the truth state of the environmental predicate or changing the state of the environment. We can also represent information about environmental processes occurring outside the robots, modeling them probabilistically using exponentially timed transitions with a certain rate λ. Memory places connected to such transitions allow modeling both the impact such processes have on the robots and the robots have on the environment.

5.4.2 IAC-based Generalized Stochastic Petri Net

Starting from the EPN structure of the IAC, we obtain a GSPN representation of that IAC that will be the macroscopic model of the system. To do so, we identify which transitions correspond to immediate or exponentially timed transitions. All control transitions added in the higher layer during IAC composition are immediate, since their associated condition in the EPN is always true. Initial and final conditions of institutions can be either immediate or exponentially timed. It is up to the designer of the model to decide, according to problem domain-knowledge, which option reflects more accurately the system being modeled. This decision must also be taken for lower layer transitions. While actions and conditions associated with places and transitions in the EPN might be useful

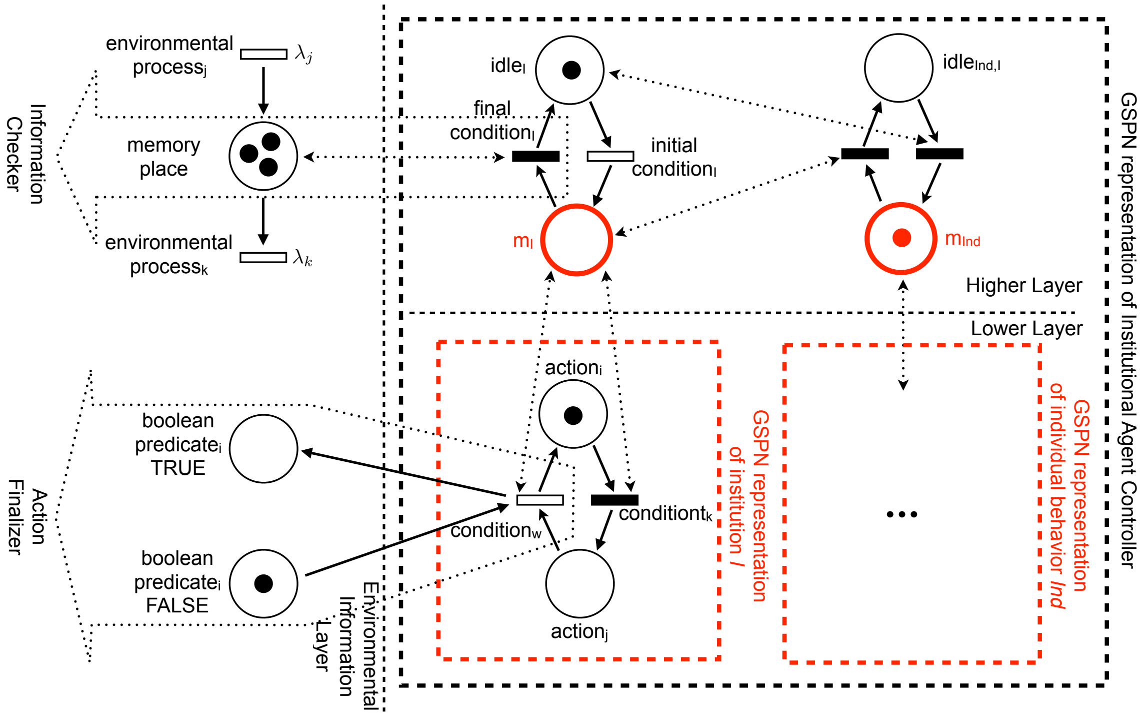


Figure 5.2: Example of the GSPN representation of an abstract IAC composed of one institution I and an individual behavior Ind. This GSPN is connected to information about the environment, present in the environmental information layer. Example of classification of transitions into *information checkers* and *actions finalizers*.

for the designer of the model to better classify transitions, the GSPN models use only the PN structure of the EPN, which includes place and transitions labels but not actions and conditions information. Fig. 5.2 displays the GSPN representation of an abstract IAC.

We can further classify the transitions in the IAC layers of the GSPN model into two classes: *action finalizers* and *information checkers*. Action finalizer transitions model the execution (and termination of such execution) of a certain action by the robot. When connected to places in the environmental information layer, such transitions change the marking of the GSPN, representing the robot action changing the modeled system. An example is the transition associated with $condition_w$ shown in Fig. 5.2. Information checker transitions connect with places in the environmental information layer using bidirectional arcs, representing that their firing does not change the state of the environment, but needs a certain set of environmental conditions to be verified to occur. An example is the transition associated with $finalcondition_I$ shown in Fig. 5.2.

Not all transitions in the model must be classified as actions finalizers or information checkers. Only transitions that need to connect to the environmental information layer need to fall into one of these two categories. It must also be noted that the diagram of Fig. 5.2 does not impose any restrictions on how the classification of transitions is performed. Both types of classification may be used for both lower layer and higher layer transitions. It is up to the designer of the model to make a correct use of such classification.

The final step in obtaining the IAC-based GSPN model is to estimate the firing rates of exponentially timed transitions. These can be estimated based on properties of the system, for instance, robots' maximum velocity, sensor noise, geometrical properties of the environment, etc. Using these estimated firing rates in the GSPN gives us an *a priori* model that can be used to generate predictions of the system before any implementation. Since the GSPN model is part of a multi-level modeling approach, we can also use data from from models at lower levels of abstraction (submicroscopic, microscopic) to estimate the firing rates. In this case, we need to have an initial implementation of the system in one of those layers.

It must be noted that not all the layers must be present in every GSPN model designed. If the IAC is sufficient to obtain accurate predictions of metrics of interest, there is no need to include dispensable environmental information. The same can be said of the IAC. If we are interested in modeling only how different behaviors relate to one another, it is possible that the implementations of those behaviors in the lower layer are not essential. If we are interested only in modeling how a particular institution performs, the higher layer composition of the IAC might not be needed. The modular design of our controller and model should serve as an advantage for the designer, allowing the selection of relevant information and the dismissal of extraneous information.

Summary

In this chapter we introduced our approach to the probabilistic modeling of distributed robotic systems controlled by IACs. We follow a MLMM that considers models in different levels of abstraction – submicroscopic, microscopic (spatial/non-spatial), macroscopic. Our approach focuses on the macroscopic level, using the IAC as a starting point for a GSPN probabilistic model. GSPNs extend PNs with the notion of timed transitions (with exponentially distributed firing delays). We discussed that the marking process of a GSPN is a semi-Markovian process and can be analyzed via an EMC. We extend the GSPN model with the introduction of an environmental information layer, where information not present in the IAC can be represented. The modular approach to controller and model design allows a designer to select relevant sections of the IAC, represent them as a GSPN, and perform steady state analysis to obtain accurate predictions of the overall performance of the system.

Chapter 6

Validation of the Control and Modeling Methodology: The Wireless Connected Swarm Case Study

In order to illustrate how to apply our methodology, obtaining an IAC that performs a desired task and allows us to build a GSPN macroscopic model of the system, we have selected a case study previously investigated in [Nembrini et al., 2002, Winfield et al., 2008], where a decentralized control algorithm is able to maintain a certain degree of spatial compactness of a robotic swarm using exclusively as information the current number of wireless connections robots have with their neighbors.

When applying our formalism, 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 behaviors. The low complexity of this case study in terms of social interactions among robots, makes it a good candidate to assess the ability of our IAC approach to replicate results obtained with other approaches. To verify this, we compare results from two submicroscopic models implemented through realistic simulations, one using the IAC and a second using the original FSA presented in [Nembrini et al., 2002].

It must be noted that the same low complexity that makes this a good case study for the validation of our methods, also makes the benefits of the institutional robotics approach not easily noticeable. Case studies with more complex tasks and scenarios of strong social nature will be introduced in the following chapters in order to fully explore such benefits.

In a second set of experiments, we verify if the submicroscopic model implemented is faithfully capturing the reality and can be used to further optimize the performances of distributed control strategies using an institutional robotics approach. To do so, we perform real world experiments, duplicating accurately the conditions used, including a large number of robots (up to 40) and noisy communication channels.

Finally, we show that a GSPN view of the IAC for this case study, in closed loop with the distributed robotic system it controls, enables quantitative model-based analysis. We use the GSPN view of the institutions designed for our case study to derive a macroscopic model that captures the mean-field dynamics of the distributed robotic system. We compare our macroscopic model predictions with results from the submicroscopic model.

In [Evans et al., 2010], authors follow the same modeling methodology, and validate a macroscopic model with real world experiments, while also performing submicroscopic simulations. With the intent of comparing performance with a deterministic and a probabilistic controller, a robotic swarm has to execute a chain formation task. Simulations are performed with *Webots*, while real world experiments are performed with 19 *Alice* robots [Caprari and Siegwart, 2005]. In works presenting both simulation and real robots the mutual interconnection role is not always the same. In [Gross et al., 2006], a self-assembly task is considered, with controllers for the robots being designed in simulation and then transferred to reality. While only 16 *s-bots* are used in reality, results are shown in simulation with up to 100 robots. Effects of sensor and actuator noise on robotic swarms performance is studied on [Hoff et al., 2011] in 3 standard tasks: trail following, swarm expansion, and line formation. Different types of sensors and actuators are compared, with results being shown both on a custom made simulator and on (at least) 10 real *e-puck* robots. Other swarm robotics studies [Guo et al., 2011, Santos and Chaimowicz, 2011] use simulations to provide important results on algorithms and techniques, but offer only proof of concept experiments with real robots (in both cases authors use *e-puck* robots).

This chapter is organized as follows. In Section 6.1 we describe the environment in which the robots operate and the task they must accomplish. Section 6.2 describes two different approaches to controller design for the task at hand, while Section 6.3 describes the performance metrics. A comparison between these two approaches in simulation is described in Section 6.4. The faithfulness of such simulations to reality is assessed in Section 6.5. Finally, in Section 6.6 our modeling methodology is applied to this case study to analyze the stochastic performance of the system.

6.1 Environment, Robots & Task Description

The original case study considered a swarm of robots located in an unbounded arena [Nembrini et al., 2002, Winfield et al., 2008]. We maintain this assumption for experiments dealing with the comparison between IAC and FSA approaches, and consider $N = 40$ robots. In a second set of experiments, we consider that the robots are located in a 3 by 3 meters arena, both in reality and in simulation. We perform experiments with $N = 20$ and $N = 40$ robots. In both cases we do not consider any other objects in the arena apart from the robots.

We use the *e-puck* robots as our robotic platform, both in reality and, through models of these robots, in *Webots* simulations. We consider two distinct communication settings for the two experiments. When comparing approaches solely in simulation, we (unrealistically) consider that the robots have perfect communication inside a perfectly circular communication radius. In the second set of experiments, real robots are equipped with a radio communication module, with bounded communication range being achieved by regulating the emission power. In simulation, we consider that communication is now simulated realistically by using a network simulation engine (OMNet++) that handles noise, fading signal propagation, channel coding, as well as a non-circular communication footprint, as a plugin for *Webots*.

The robots' task is simply to keep the swarm *coherent*. The swarm is said to be coherent if any break in its overall connectivity (robots' current number of wireless connections to neighbors) lasts less than a given time constant. A coherent swarm is considered aggregated and forms an *ad hoc* communication network between the robots.

6.2 Controller Design

We will now introduce both approaches to controller design we will be tackling. First, we introduce the FSA approach utilized in the original case study. Next, we will introduce our IAC approach.

Both approaches have points in common: we assume that X represents the number of connections perceived by the robot at any time step, and that robots send messages with their own ID every T_s time steps and update X every T_r time steps. In both approaches obstacle avoidance is implemented using a simple Braitenberg control law [Braitenberg, 1984]. Changes in the state of the controller will be linked with the α parameter, that represents the minimum number of connections a robot should have, as we will see below.

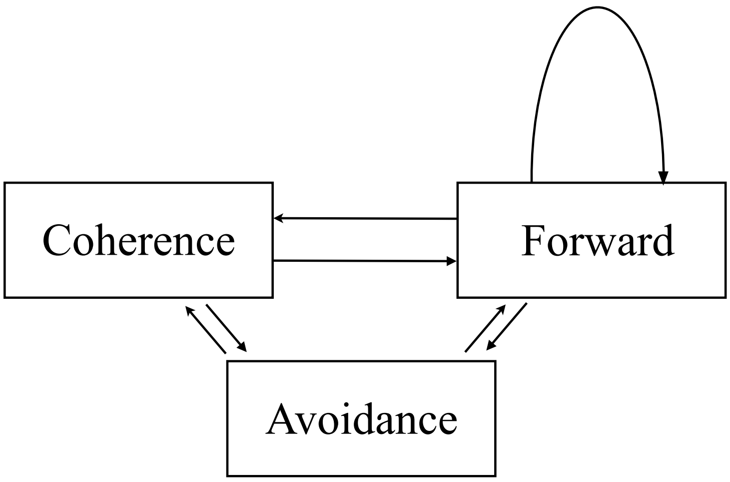


Figure 6.1: Original FSA controller for robots in the wireless connected swarm case study, with states $forward$, *coherence* and *avoidance*.

6.2.1 Finite State Automata

In Fig. 6.1 we display the FSA controller for robots in the wireless connected swarm case study, originally presented in [Winfield et al., 2008]. In the default state, defined as $forward$, the robot simply moves forward. If at any time the robot senses the loss of a connection (or more) and X falls below the threshold α (where $\alpha \in \{0, \ldots, N-1\}$), the robot assumes it is going in the wrong direction and switches to state *coherence*. In this state the robot performs a 180° turn in order to recover the lost connection(s). Upon recovering the lost connection(s), the robot performs a random turn and moves back to the $forward$ state. If the connection is not recovered, the robot simply moves to the $forward$ state. If an obstacle is detected, the robot immediately switches to state *avoidance*, where it performs obstacle avoidance for a given number of time steps, after which it returns to its previous state.

The algorithm represented with this FSA controller is called the α-algorithm. While this simple algorithm has limited robustness, it allows the swarm to maintain its connectivity to a certain extent, with its spatial compactness being controlled by the communication range and by the threshold α.

6.2.2 Institutional Agent Controller

In our IAC implementation, robots execute an individual behavior $IndAv$ (*Individual Avoidance*) and two institutions $T180$ (*Turn 180 degrees*) and TR (*Turn Random*), all

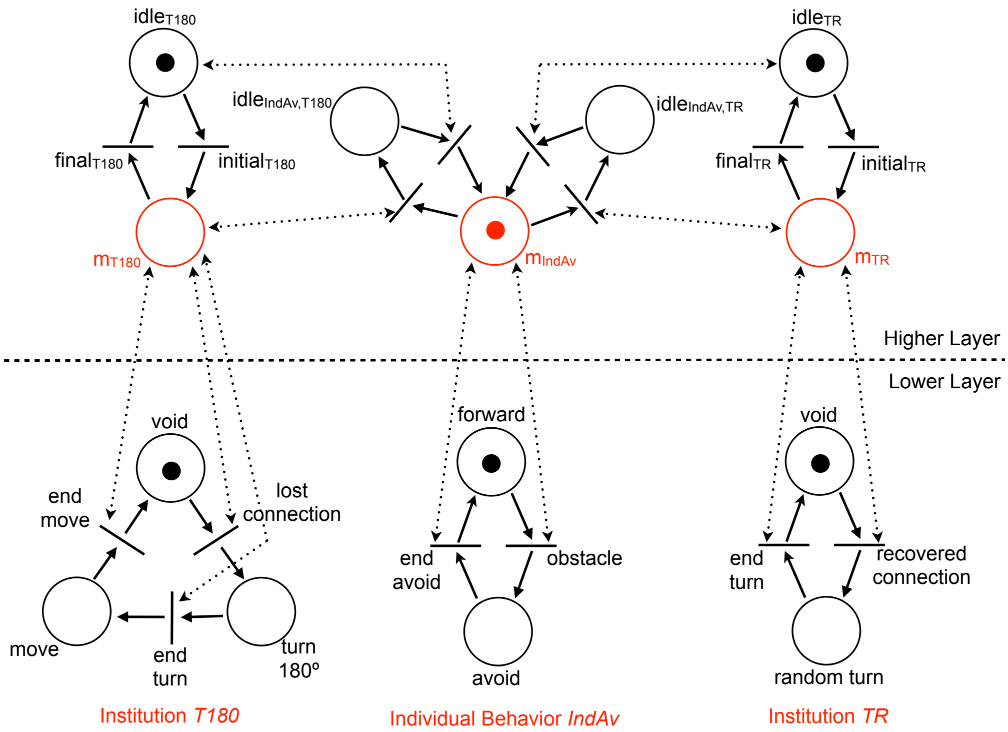


Figure 6.2: IAC for robots in the wireless connected swarm case study: EPNs for individual behavior $IndAv$ and institutions $T180$ and TR. Higher layer: composition of individual behavior and institutions.

specified by EPNs shown in the lower layer of Fig. 6.2. Individual behavior $IndAv$ specifies a behavior relating the robot to its environment. The robot moves forward and, upon detection of an obstacle, performs obstacle avoidance for a given number of time steps, after which it returns to moving forward. We do not consider this to be part of a behavior with a social nature since robots do not act upon information provided by other robots and their actions are not directed to any common goal of the swarm. Institutions $T180$ and TR implement the social rules, dealing with loss and recovery of connections. $T180$ specifies that upon losing a connection the robot performs a 180° turn followed by moving forward for a small number of steps. This short forward motion allows the robot to assess if the lost connection was recovered immediately after the execution of $T180$. Institution TR specifies that if a connection is recovered the robot performs a random degree turn.

To consider institutions as defined in Section 4.4, we need initial and final conditions and deontic operators. For institution $T180$ we say that initial condition $initial_{T180}$ is "loss of connection detected and number of connections is less than α" and the final condition $final_{T180}$ is "move forward action has ended". For institution TR we say that initial condition $initial_{TR}$ is "recovery of connection detected and previous number of connections is less than α" and the final condition $final_{TR}$ is "random turn action has ended". The deontic operator associated with both institutions is $StopInd$, specifying that institutions and individual behavior cannot be executed concurrently. We could consider $StopAll$ as the deontic operator, but due to the initial and final conditions of the institutions we know they will not be active at the same time, thus $StopInd$ is sufficient and simplifies the IAC.

We now have all the elements needed to obtain the IAC that specifies our desired behavior. The composition of the individual behavior $IndAv$ and institutions $T180$ and TR (specified separately by EPNs shown in the lower layer of Fig. 6.2) is shown in the higher layer of Fig. 6.2. The final controller is the full EPN of Fig. 6.2, obtained after merging the two layers. Both composition and merger are performed algorithmically by Algorithm 2 without the need of further designer input. Lower layer actions and 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 an FSA actions associated with states are executed.

The original three state FSA is clearly of lower complexity than the final IAC for this case study. Nevertheless, this case study does not fully capture the intended level of complexity we envision for IR applications. We choose a simple case study in order to derive sound and clear conclusions for our conjectures before moving to more complex scenarios.

6.3 Metrics

In this work, we are interested in three main metrics that represent and allow us to analyze different aspects of the swarm (and individual) behavior: connectivity, dispersion, and displacement. Of these, connectivity is the most important and the only one also present in [Winfield et al., 2008].

Definition 15 Connectivity *tells us, in average, how many robots have a particular number of wireless connections during the time needed to perform a run of the experiment.*

To measure connectivity we use data gathered by the robots about the number of time steps spent with each number of connections. Robots with α or more connections are not concerned with recovering lost connections and are likely to be moving away from the swarm. On the other hand, robots with less than α connections are actively trying to regain connections and are likely to be moving towards the swarm. Thus, we can expect the swarm connectivity to peak at α, i.e., at each time step we will have more robots with α connections than with any other number of connections.

Definition 16 Dispersion *measures the average distance of robots to the swarm center of mass.*

Dispersion gives us an indication of how spread out the swarm is across the arena. Ideally we would like this value to be as close to zero as possible, bounded by the communication radius, and constant throughout the run. To compute dispersion we use data about the position of robots gathered either in simulation or through *SwisTrack* (see Section 2.3) in the real world experiments.

Definition 17 Displacement *measures the distance between the swarm center of mass and the center of the arena.*

Given the stochastic nature of the movement of the robots, displacement will start close to zero (runs start with robots gathered closely in the center of the arena) and will increase throughout the run. The motion of the swarm as a whole resembles a random walk through the arena. This metric would be somewhat different if considered in the original case study, given that an unbounded arena was considered.

6.4 Comparing Controller Approaches

6.4.1 Experimental Setup

One of our goals is to assess the ability of our IAC approach to replicate results obtained with the original FSA approach. In order to do this, we performed submicroscopic simulations with both controllers described previously. We replicated the conditions of the original case study presented in [Winfield et al., 2008], considering $N = 40$ robots in an unbounded arena performing the task over $T = 10000$ seconds. Robots are initially aggregated in the "center" of the arena.

The connection threshold was fixed to one single value, $\alpha = 15$, and the communication radius of the *e-puck* set to 0.7 m, instead of the original 2.0 m, in order to keep the ratio presented in the original paper between communication and physical radius of the robots.

To compare the performance of the original FSA controller and our IAC approach we performed 100 runs in simulation for each implementation. Following a stability study for the task presented in [Winfield et al., 2008], we chose to gather data only after 2000 seconds of simulation on each run and average them over the remaining 8000 seconds. This allowed us to avoid storing data from an initial period of the task that might not represent a steady state of the system. During simulations we stored the number of time steps spent in each state of the original FSA controller ($forward$, *coherence*, and *avoidance*) for each number of connections (between 0 and N).

For the IAC simulations we must take into account that these states do not correspond to a single marked or non-marked place on the EPN but rather to one or more markings of the full net. For instance, state *avoidance* corresponds to markings where the individual behavior $IndAv$ is active (macro place m_{IndAv} marked) and the *avoid* place is also marked. State *coherence* corresponds to all markings where institution $T180$ is active (macro place m_{T180} marked). All other markings correspond to state $forward$.

6.4.2 Results and Discussion

In Fig. 6.3 and Fig. 6.4 we present the connectivity metric results for $N = 40$ robots following the FSA approach and the IAC approach, respectively. The results are averaged over the $N = 40$ robots and all runs. Results for state *coherence* are only available for connectivities below α (no robots in *coherence* have more than α connections). The dotted line (small round dots) corresponds to the sum of robots in all states. Results show a good agreement between the two approaches, both in number of robots and state distribution for each number of connections. The small differences between results are due to a small difference in the way both controllers update the neighborhood information. Results also show a good agreement with those presented in the original case study [Winfield et al., 2008].

6.5 Comparing Submicroscopic Model and Reality

6.5.1 Experimental Setup

In order to verify if the submicroscopic model implemented was faithfully capturing the reality we performed a different set of experiments (both real robot experiments and *Webots* simulations) with sets of $N = 20$ and $N = 40$ robots in a 3 by 3 meters arena performing the task over $T = 1800$ seconds (it would be highly unpractical to consider $T = 10000$ as before). Again, robots are initially aggregated in the center of the arena.

The connection threshold is dependent on the size of N and is set to 40% of N, $\alpha = 8$ for $N = 20$ and $\alpha = 16$ for $N = 40$. We set the transmission power of the *e-puck* communication module to an appropriate value that allows us to roughly achieve the desired communication radius.

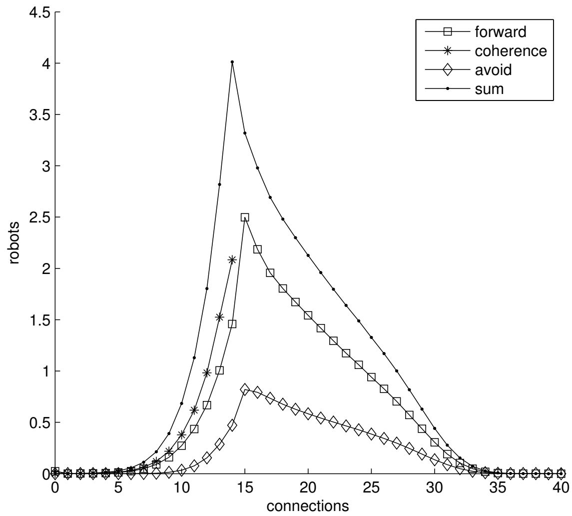


Figure 6.3: Connectivity metric: average number of robots in each state (and sum of all states) with a particular number of connections for $N = 40$ simulated robots following the FSA approach ($\alpha = 15$).

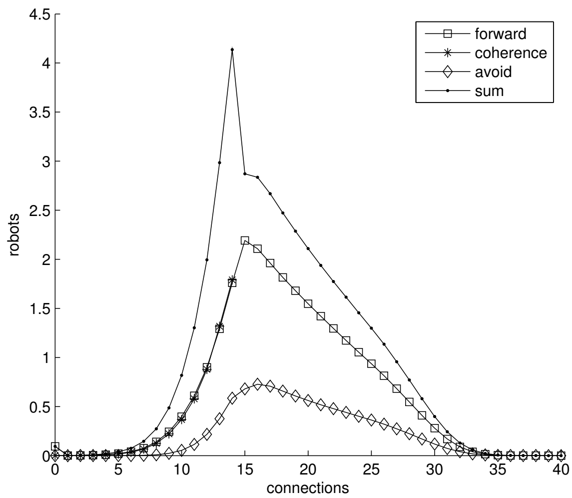
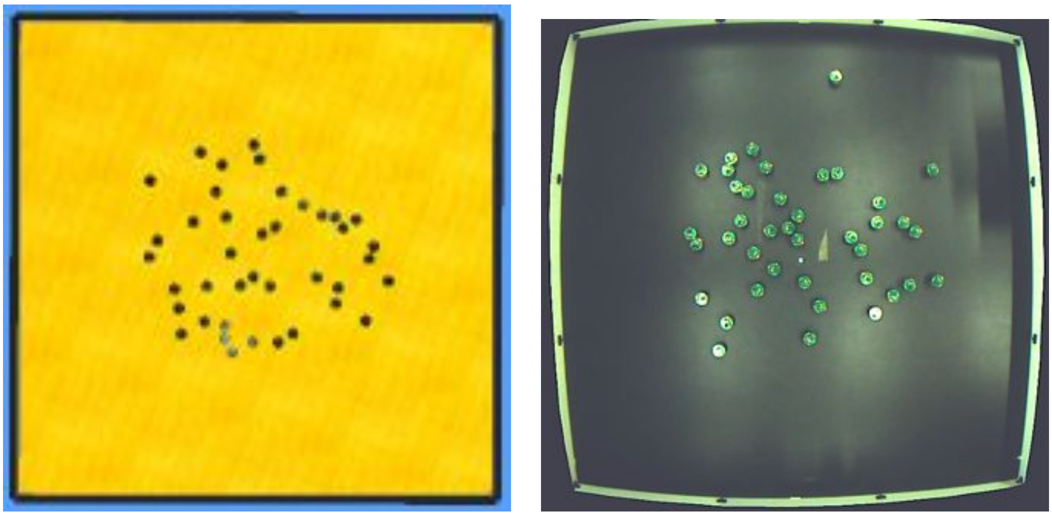


Figure 6.4: Connectivity metric: average number of robots in each state (and sum of all states) with a particular number of connections for $N = 40$ simulated robots following the IAC approach ($\alpha = 15$).



(a) (b)

Figure 6.5: (a) *Webots* simulation screenshot, 40 *e-puck* robots simulated. (b) Real world experiment screenshot, 40 *e-puck* robots.

To compare the performance of our submicroscopic model and real world experiments we performed 100 runs in simulation for each $N = 20$ and $N = 40$, and 10 runs in reality for $N = 20$ and 5 runs for $N = 40$. During runs we stored the number of time steps robots spent with each number of connections (between 0 and N), but not the state of the controller. We also recorded videos of the arena during the real world experiments using an overhead camera and the *SwisTrack* software. We processed the videos offline, using *SwisTrack* to perform background subtractions and blob detection, in order to extract and store the position of each robot in each frame. We also stored information about the position of robots at each time step of our simulations. Fig. 6.5 displays a screenshot of a simulation and an image of the arena during execution taken with the overhead camera.

Videos of the real robot experiments can be found here (for $N = 20$)[1] and here (for $N = 40$)[2]. Videos of the *Webots* simulations can be found here (for $N = 20$)[3] and here (for $N = 40$)[4].

[1] See http://www.youtube.com/watch?v=k3Wya6ty63w.
[2] See http://www.youtube.com/watch?v=zKw1bwccnKA.
[3] See http://www.youtube.com/watch?v=fnkQfV0P1SY.
[4] See http://www.youtube.com/watch?v=zfokvHcQD_U.

6.5.2 Results and Discussion

In Fig. 6.6 and Fig. 6.7 we present the connectivity metric results for $N = 20$ and $N = 40$. In green we display results obtained with the submicroscopic model, while in red and blue we display results with real robots. The blue line was obtained with data about number of connections as perceived and recorded by the robots. On the other hand, the red line was obtained offline using *SwisTrack* by counting, for each robot, how many other robots were present in a 0.7 meters range, somehow emulating a perfectly radial communication disk. The differences in these two lines can be explained by the spatially irregular coverage of the wireless radio communications. The blue line reflects more accurately this noisy nature by spreading the number of robots more evenly between 3 and 9 connections in Fig. 6.6 and producing a second local maximum for 4 connections in Fig. 6.7. This maximum can be explained by the increase in N and α. The increase in α forces robots to try to keep more neighbors in their communication radius, leading to robots aggregating in a smaller space. This effect is magnified by the increase of robots in the swarm. Thus, when robots lose or gain connections they lose 1 or 2 connections with $N = 20$ but they lose 4 or 5 connections with $N = 40$. The video data processed with *SwisTrack* always gives the correct number of neighbors since all robot positions are known, thus the red line better reflects the overall swarm behavior. We can see that connectivity measured with *SwisTrack* has a very good agreement with the connectivity measurements obtained in our submicroscopic model. The slight shift of the curve of the simulations in relation to the curve of *SwisTrack*, representing that robots have in average slightly less connections, is most likely a product of the inclusion of wireless communication realism (noisy fading and ellipsoidal communication area) in the simulations through the OmNET++ plugin. These results also show a very good agreement with the results presented in the original case study work [Winfield et al., 2008].

Fig. 6.8 and Fig. 6.9 display the dispersion metric results. Real robots results are obtained only using the video data processed with *SwisTrack*, since robots do not have localization capabilities and are unaware of their own location as well as the location of others. We can see that despite a small difference between real robots and the submicroscopic model, the results still show a good agreement. As expected, the distance to the swarm center of mass is close to zero, smaller than the communication radius (0.7 meters) and constant (within some bounds) throughout the run. The small variations in this distance indicate an expansion and contraction motion performed by the swarm while losing and consequently trying to regain connections. This can be observed mainly in the real robots results, since the elevated number of runs performed in simulation diminishes the effect.

In Fig. 6.10 and Fig. 6.11 we present the displacement metric results for $N = 20$ and $N = 40$. Again, real robots results are obtained only using the video data processed with *SwisTrack*, for the reasons previously stated. As expected, displacement distance

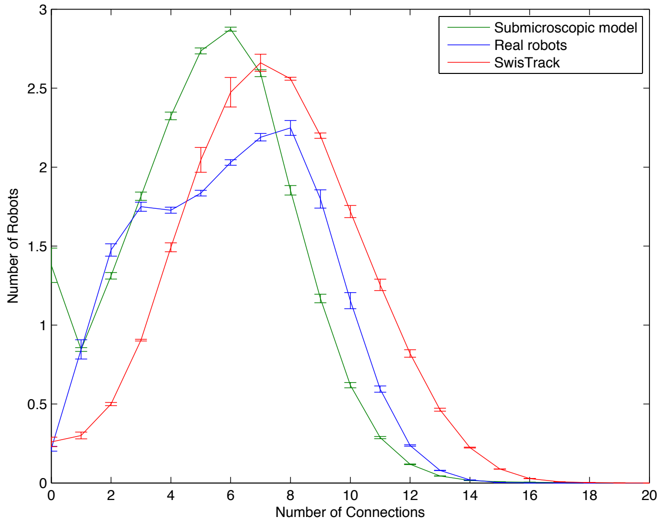


Figure 6.6: Connectivity metric: average number of robots with a particular number of connections during a run. Variance shown for different runs. Results for $N = 20$ robots and $\alpha = 8$.

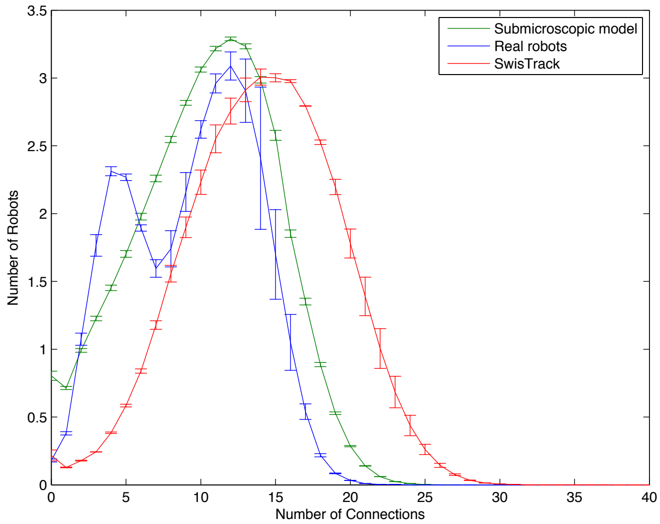


Figure 6.7: Connectivity metric: average number of robots with a particular number of connections during a run. Variance shown for different runs. Results for $N = 40$ robots and $\alpha = 16$.

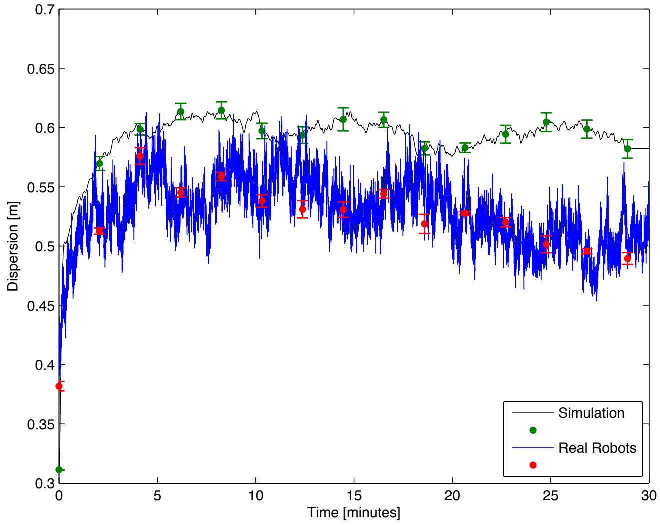


Figure 6.8: Dispersion metric: average distance of robots to swarm center of mass throughout a run. Green and red dots display variance for simulation and real robots results, respectively. Results for $N = 20$ robots and $\alpha = 8$.

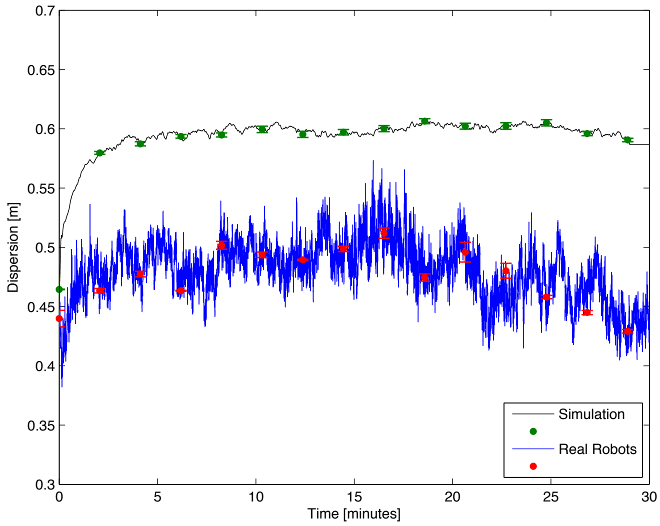


Figure 6.9: Dispersion metric: average distance of robots to swarm center of mass throughout a run. Green and red dots display variance for simulation and real robots results, respectively. Results for $N = 40$ robots and $\alpha = 16$.

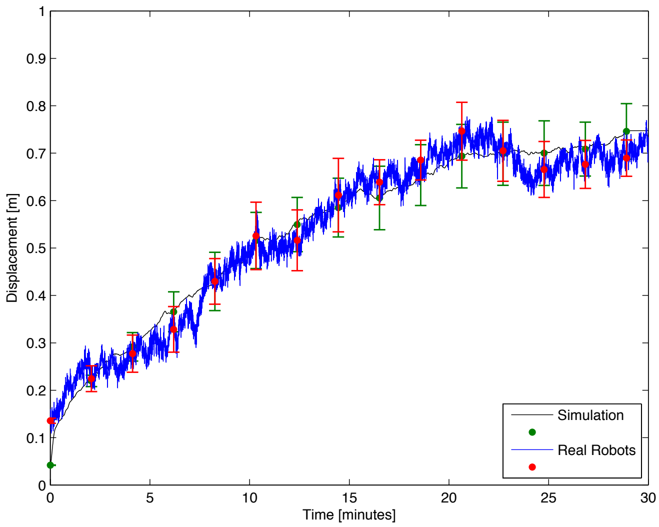


Figure 6.10: Displacement metric: average distance of swarm center of mass to arena center throughout a run. Green and red dots display variance for simulation and real robots results, respectively. Results for $N = 20$ robots and $\alpha = 8$.

is close to zero at the beginning and increases throughout the run. For $N = 20$, the submicroscopic model and real robot experiments show perfect agreement. However, for $N = 40$, despite distance increasing in both simulation and real robots, we observe that the rate of increase is doubled from simulation to real robots. A possible explanation for this effect is the difference in the obstacle avoidance behavior. While in submicroscopic simulations *e-pucks* are considered as perfect cylindrical blocks, in reality *e-pucks*' bodies are translucent. This leads to some collisions between robots, being this effect greatly increased when the number of robots is doubled and they are forced to aggregate in a smaller space (because α is also doubled). Robots motion becomes less predictable and more stochastic and as a result the displacement of the whole swarm is increased, much in the same manner as a random walk with increased turning probability. This difference also helps explain the slightly worst matching (with respect to Fig. 6.8) between reality and simulation in Fig. 6.9.

6.6 Modeling State Distribution with Institutional Agent Controller

In this chapter we are not concerned with modeling all details of our case study. Our goal is to establish that IACs can be used in modeling by providing the necessary structure to

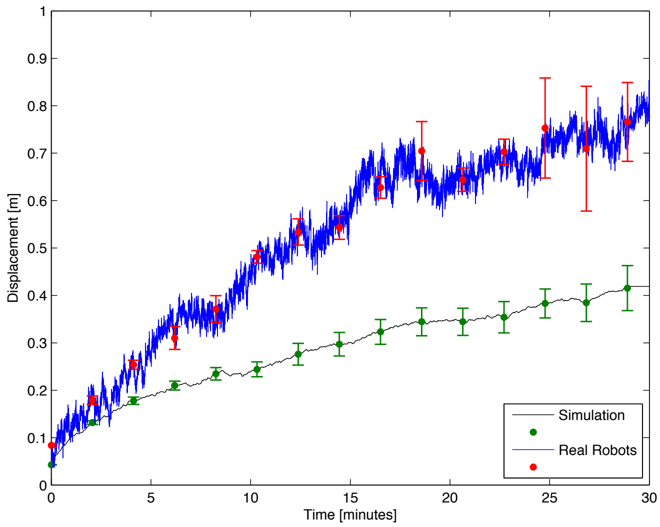


Figure 6.11: Displacement metric: average distance of swarm center of mass to arena center throughout a run. Green and red dots display variance for simulation and real robots results, respectively. Results for $N = 40$ robots and $\alpha = 16$.

construct the desired GSPN models. Our approach is to use directly the EPN structure of the IAC to build our GSPN macroscopic model. We focus on steady state analysis to assess the quality of the model. To do so, we use the software PIPE2 (described in Chapter 2) both for editing and performing steady state analysis of our models, based on the EMC analysis described in Section 5.3.

In [Winfield et al., 2008], the MLMM was applied to the wireless connected swarm case study. The FSA controller is used as a starting point for a Probabilistic FSA (PFSA) model that describes the swarm connectivity and overall state distribution of the swarm. In this work, we will focus on modeling the overall state distribution over the possible states, i.e., the average probability of a robot being in each state ($forward$, $coherence$, and $avoid$).

6.6.1 Model Structure

As discussed, by using the two layers of the IAC we can select sections of the EPN that are of interest to model a given system. For our proposed model, the higher layer would suffice if only states $forward$ and $coherence$ were considered (with $coherence$ corresponding to markings where the macro place for institution $T180$ is marked). However, to also consider state $avoid$ we need to include the lower layer implementation of the individual behavior, in order to make the distinction between $avoid$ and $forward$. We use the

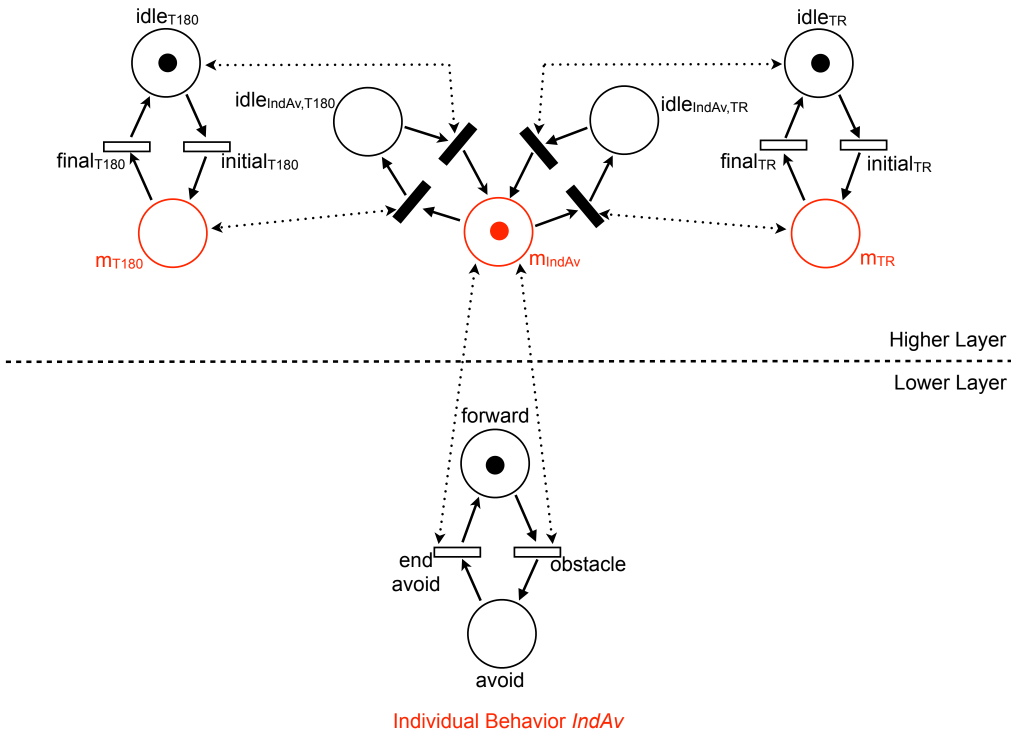


Figure 6.12: IAC-based GSPN probabilistic macroscopic structure. Immediate transitions represented as full boxes, timed transitions represented as empty boxes.

association between markings of the IAC and states of the original FSA as described in Subsection 6.4.1. The IAC-based GSPN structure for our macroscopic model is presented in Fig. 6.12.

The only immediate transitions in the model are those that are not associated with any condition. These correspond to the control transitions added during composition of behaviors, and are the transitions linking the macro place of the individual behavior m_{IndAv} with idle places $idle_{IndAv,T180}$ and $idle_{IndAv,TR}$. The remaining transitions in the model are stochastically timed and their transition rates need to be estimated. These correspond to conditions *obstacle*, *end avoid*, and the initial and final conditions for both institutions.

6.6.2 Rates Estimation

Since our goal is only to establish that the IAC structure can be used as a GSPN macroscopic model, we choose to estimate the transition rates directly from data gathered during the submicroscopic simulations performed when comparing controller approaches. The transition rates $\lambda_{i,k}$ are calculated separately for each timed transition i and each number of connections ($k = 0, \ldots, 40$) following:

$$\overline{t_{i,k}} = \frac{t_{i,k}}{N * N_{Runs} * T_{TS}}, \quad \overline{f_{i,k}} = \frac{f_{i,k}}{N * N_{Runs} * T_{TS}}, \quad \lambda_{i,k} = \frac{\overline{f_{i,k}}}{\overline{t_{i,k}}} \tag{6.1}$$

where $t_{i,k}$ is the number of total number of time steps the input places of transition i are marked and $f_{i,k}$ is the total number of times transition i fires, counted for all robots in all runs. T_{TS} is the number of time steps during one run, the number of robots is $N = 40$, and the number of runs is $N_{Runs} = 100$.

6.6.3 Results and Discussion

For each number of connections k, we perform steady state analysis on our GSPN macroscopic model with rates $\lambda_{i,k}$ for transitions *obstacle, end avoid*, $initial_{T180}$, $final_{T180}$, $initial_{TR}$ and $final_{TR}$. This analysis gives us the steady state probability for each tangible marking in our GSPN model. The desired state distribution model is obtained by adding the probabilities for all markings corresponding to each state of the original FSA. Note that, for each number of connections k, this model actually gives us a probability mass function for the discrete random variable describing possible states of the robots.

The macroscopic model's results are displayed in Fig. 6.13. They show how the probability of being in each state varies with the number of connections k. In Fig. 6.14, we display the state distribution based on data acquired with the submicroscopic model (connectivity results displayed in Fig. 6.4), obtained by dividing the number of robots in each state by the total number of robots, for each number of connections k. In Fig. 6.15 we display the Kullback-Leibler (KL) divergence between the probability mass functions given by the macroscopic model (for each k) and the implicit probability distributions (for each k) given by the data gathered using the submicroscopic model. KL divergence between distributions p and q effectively measures the average likelihood of observing data with the distribution p if the particular model q actually generated the data [Cover and Thomas, 2006].

We can observe an almost perfect matching between the two results. This is somewhat expected, since our estimation of transition rates comes directly from the data gathered during the submicroscopic simulations. Nevertheless, it was not obvious that the probabilistic distribution of firing delays of transitions is exponential. The larger error in Fig. 6.15 comes from the extremely low number of time steps and transition fires for robots

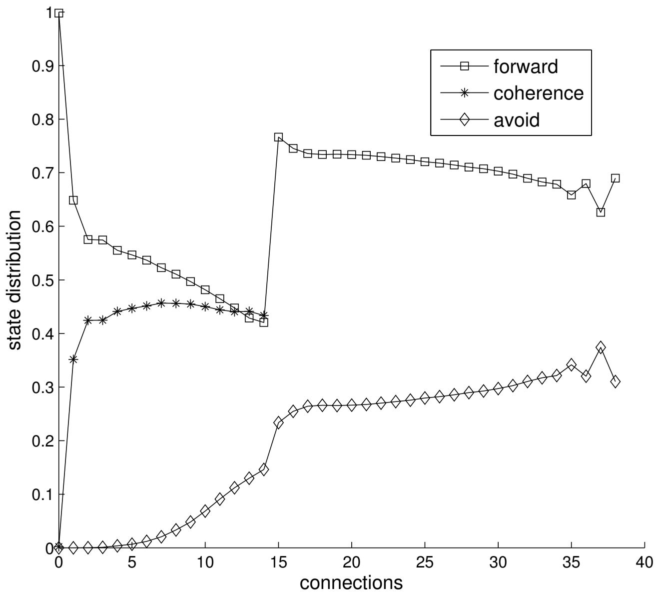


Figure 6.13: State distribution predicted from GSPN macroscopic model.

with 38 connections. This affects the calculation of the correct rate and thus generates a bigger error in the model. We conclude that with a correct estimation of transition rates, our IAC provides a good structure for the generation of macroscopic models.

Summary

In this chapter we focused on assessing our proposed methodology. We implemented two submicroscopic models of the wireless connected swarm case study through realistic simulations. The original version of the case study uses an FSA controller. The second version uses an IAC composed of one individual behavior and two institutions designed to perform the task similarly. Results from submicroscopic models show a good agreement between the two versions. These submicroscopic models were validated by implementing the case study with a swarm of (up to) 40 real, resource-constrained robots. We also assessed the quality of a macroscopic model of the case study, obtained with our IAC-based hierarchical methodology. Using the EPN structure of the IAC designed for the case study, we were able to construct a GSPN macroscopic model for the overall state distribution of the system. By using data gathered from submicroscopic simulations in order to estimate the transition rates necessary for our GSPN macroscopic model, we were able to observe a very good agreement between macroscopic and submicroscopic model's predictions.

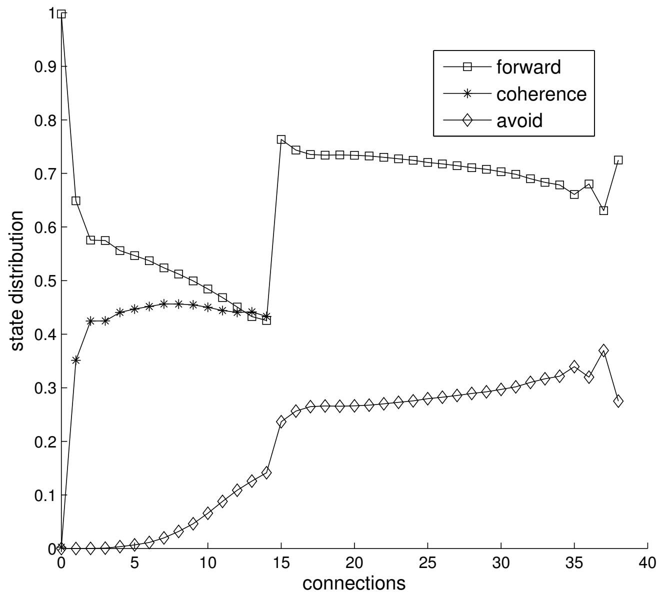


Figure 6.14: State distribution from the submicroscopic model.

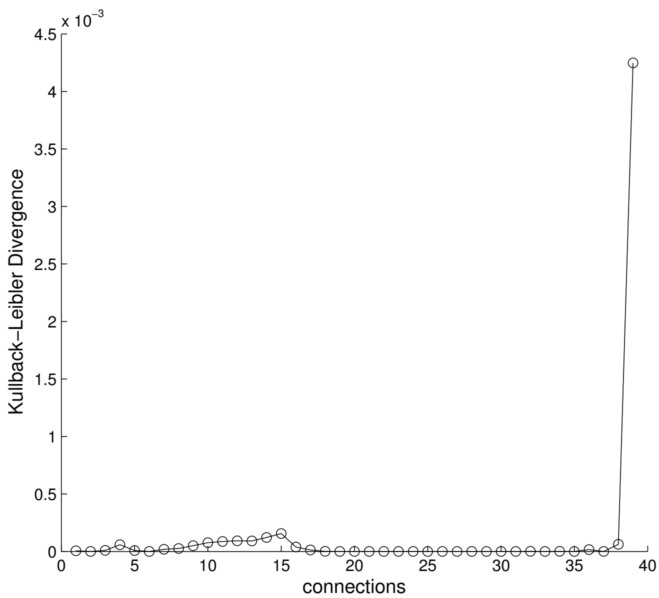


Figure 6.15: Kullback-Leibler divergence (for each k) between the probability mass functions given by the GSPN macroscopic model (for each k) and the implicit probability distributions (for each k) given by the data gathered from submicroscopic simulations.

Part III

Institutional Robotics Case Studies

Chapter 7

Coordination Through Institutional Roles: The Corridor Case Study

As discussed in previous chapters, institutions can take several forms: organizations, norms, hierarchies, roles, etc. In the wireless connected swarm case study, we experimented with institutional norms in a low complexity task. We now focus on another important institutional form, the institutional role. Our objective is to observe if coordination of a distributed robotic system can be improved by using institutional roles, possibly in combination with other types of institutions. Also, the task to be accomplished by the robots in this case study is of higher complexity (w.r.t. the wireless connected swarm case study), with complexity of social interactions among robots depending on the approach taken.

We take an institutional role as a behavior specific to a subset of all robots, that can be seen (by an internal or an external observer) as functional to some collective task or activity, and that depends on other robots' behaviors (in the sense that others must recognize and/or permit such role playing by particular robots). Important related issues, that require conceptual and experimental refinement, and partly addressed by the experimental work reported in this section, are role assignment (how some robots start playing a role), role recognition (how robots recognize that some others are playing a role), and role permission (how robots permit other robots to play a role and behave accordingly).

In this case study, we demonstrate a concrete example of how the concept of institutional roles can advance coordination strategies and improve the task performance of a robot team. We use the following scenario: a team of robots is situated in an environment containing two rooms connected by a narrow corridor. The robots must continuously transport virtual items between the two rooms. The corridor is too narrow for two robots moving in opposite directions to pass one another. In order to avoid congestion in the corridor, the traffic between the two rooms must to be regulated so that robots only attempt to traverse the corridor in one direction at a time.

We show how such regulation can be done when the constituent robots are given the capacity to assume an institutional role, that of traffic regulator. We compare the institutional approach to a self-organized approach to the same task and try to identify in which situations one is preferable to the other. In the last section, we exemplify how institutions can be used in collective decision-making processes.

It must be noted that this case study was initially developed before the IAC formalism was introduced, as an early effort to observe the potential impact of the institutional robotics approach. In the next sections we present a IAC for this case study obtained with the methods described in Chapter 4. However, all the experiments were performed using a more traditional FSA approach also presented. This case study was implemented solely using submicroscopic simulations due to hardware constraints (more details in the next section).

This chapter is organized as follows. Section 7.1 describes the environment in which the robots operate, and the task they must accomplish. Sections 7.2 and 7.3 present two different approaches to the case study, with controller design being discussed in Section 7.4. In Section 7.5 we discuss the concept of institutional roles in more detail. Experimental setup, metrics, and results, for this case study are presented in Sections 7.6, 7.7, and 7.8, respectively. A further study on collective decision-making within this case study is presented in Section 7.9. This study provides a relevant first step towards autonomous adaptation of institutions.

7.1 Environment, Robots and Task Description

As with the previous case study, we use the *e-puck* robots as our robotic platform. We make use of the robots' proximity sensors, camera, and differential drive system. We also consider that each robots is endowed with two different forms of communication, short-range and long-range. Long-range communication can be achieved using Bluetooth, while short-range communication can be achieved using the *e-puck* proximity sensors as an infrared communication device. Unfortunately, using the proximity sensors in such manner, simultaneously with the camera, creates conflicts in the interrupt structure of the *e-puck*'s microcontroller, preventing us from implementing this case study with the available firmware.

Robots are situated in an arena consisting in two rooms connected by a narrow corridor (see Fig. 7.1). The width of the corridor allows only for one robot. In the left room robots can pick up "virtual payloads" (in infinite supply) that can be deployed in the

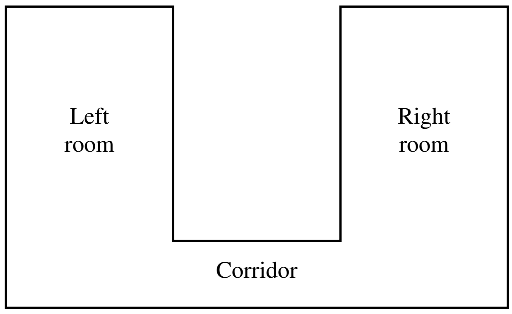


Figure 7.1: The arena.

right room. Both picking up and deploying virtual payloads happen after a fixed amount of time has elapsed since the robots enter the respective rooms. In order for the robots to recognize their location, the walls have different colors, yellow in the left room, green in the right room and blue in the corridor.

The team goal is to maximize the number of deployed virtual payloads. 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. 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 in some way to avoid deadlock situations in the corridor.

7.2 Institutional Approach

We designate by *transporting* robots all robots that are transporting virtual payloads, and thus, actively accomplishing the task. Robots performing an institutional role are designated as *regulators* (we use also *traffic regulators* interchangeably).

7.2.1 Transporting Robots

Initially, all robots are transporting robots. They are placed randomly in the two rooms (see Fig. 7.2-(a)) where they attempt to locate a wall and perform a wall-following behavior by keeping a wall on their right hand side, using readings from their proximity sensors. This wall-following behavior is complemented with some use of the camera in order to avoid conflicts with other transporting robots. If an obstacle is detected by any of the front proximity sensors, the robot captures an image with its camera. Based on the colors detected in the image, the robot can distinguish between other robots and walls. In case the obstacle is a wall, the robot turns and follows the newly encountered wall. If the robot determines that the detected obstacle is another robot, it decreases its speed (and stops if necessary) in order to avoid collisions.

Robots also use their camera when, based on the readings from their proximity sensors, there is the possibility that they might be entering or leaving the corridor. If a robot, using its proximity sensors, registers nearby obstacles or walls on both its left hand side and on its right hand side, it captures an image in order to determine if it is in the corridor or not. If the robot has previously determined that it is in the corridor and if a proximity sensor on either the left hand side or on the right hand side stops to register nearby obstacles or walls, the robot captures an image again to determine if it has exited the corridor or not.

By navigating in this manner through the arena, robots are able to pick up virtual payloads and deploy them.

7.2.2 Traffic Regulation

If the need for traffic regulation arises due to a conflict between two transporting robots in the corridor, two robots assume the institutional role of traffic regulators. The two traffic regulators place themselves at the opposite ends the corridor so that each regulator can control the flow of transporting robots entering the corridor from one of the rooms (see Fig. 7.2-(b)). 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 one time. The synchronization between the regulators is facilitated by an external program running on a *Webots* supervisor node. The traffic regulators use long range communications to interact with the supervisor. The use of the supervisor is instrumental to our specific testbed. Synchronization between regulators could be designed in a decentralized manner, since the robots are able to communicate with one another over long distances. We did not follow this option because our real world implementation of long rage communication used Bluetooth, and *e-pucks* could only use this technology as a client, thus preventing communication via Bluetooth between *e-pucks*.

The regulators use their short range communication capabilities to emit messages to guide the transporting robots trying to enter the corridor. A traffic regulator periodically emits messages when it has to prevent transporting robots from entering the corridor from the room in which it is placed. Transporting robots have to be inside the short range communication radius of the regulators (set to 15 cm) to receive messages. If a transporting robot receives a message to stop, it will stop and begin to relay the stop message so other transporting robots behind it will stop too. As a result, the transporting

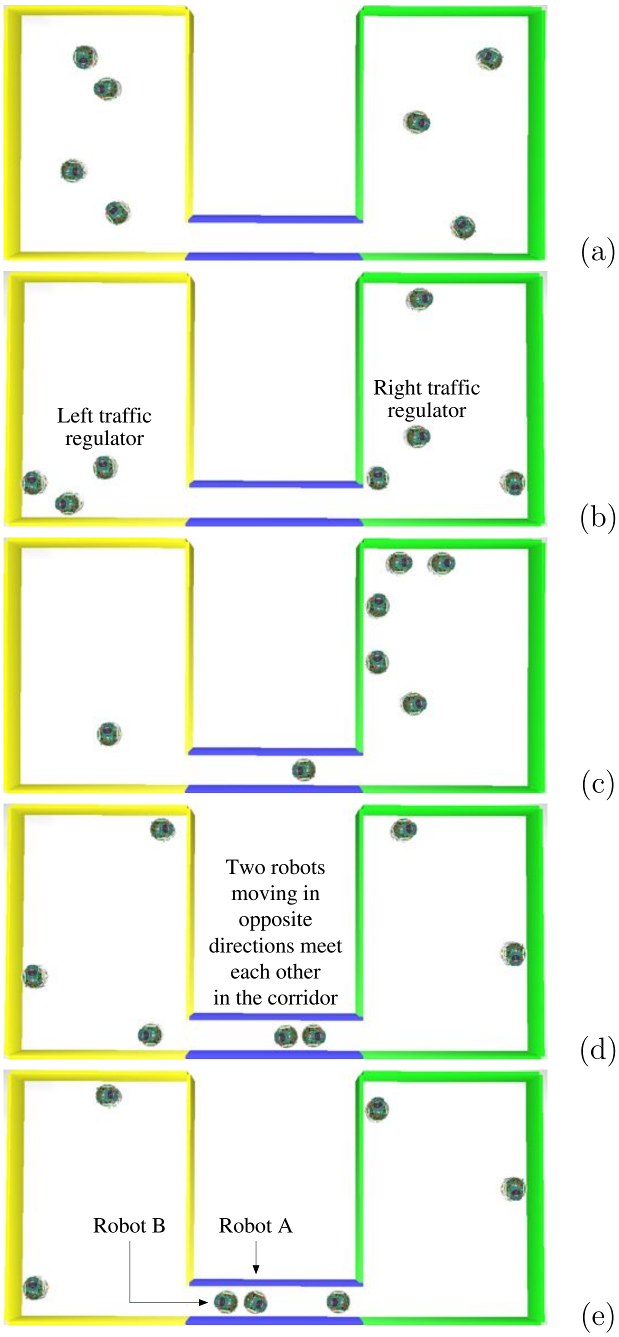


Figure 7.2: Screenshots from submicroscopic simulations: (a) initial deployment of robots in the rooms; (b) regulators in their final positions at each entrance of the corridor; (c) queue formed behind right traffic regulator, while robot moves in the corridor. (d) two robots encounter each other in the corridor; (e) after adopting the role, robot A switches the role with robot B.

robots will form a queue (see Fig. 7.2-(c)). When the first robot in the queue receives a message to proceed, it forwards the message to any robots that may be behind it, and the queued up robots will start to move.

7.2.3 Allocation of the Traffic Regulator Role

When two robots moving in opposite directions encounter one another in the corridor (see Fig. 7.2-(d)), they send a message to the supervisor to determine if they should adopt the role as traffic regulators. Each of the robots specifies from which room it came. If no other robot has yet assumed the role in the room specified by the robot, the supervisor instructs the robot to assume the traffic regulator role in that room. The robot, now that it has adopted the role, has to retreat to the room from which it came and place itself next to the entrance of the corridor. However, after two robots moving in opposite directions have assumed the role as regulators, other robots may already have entered the corridor and prevent them from navigating to the entrance of the corridor (see Fig. 7.2-(e)). In order to speed up conflict resolution in this case, the role is propagated to the last robot that entered the corridor from a given direction. Role propagation takes place in the following way: a traffic regulator (robot A) has been assigned the role, but has not yet navigated to the right location. During its retreat, robot A encounters another robot (robot B) in the corridor, both robots detect one another. Robot A stops, while robot B immediately sends a message to the supervisor in order to discover if it should assume the role as a traffic regulator. Despite the fact that a traffic regulator has already been assigned to the room from which robot B came (namely robot A), the regulator it is still located inside the corridor and not yet coordinating traffic. Thus, the supervisor sends a message to robot A to cancel the role assignment and instructs robot B to adopt the role instead. Robot A abandons the role, turns around and assumes the behavior of a regular transport robot.

After exiting the corridor, the regulator sends a message to the supervisor stating that it has made it outside of the corridor, preventing the supervisor from propagating the role further. The regulator navigates to a specific position at the entrance of corridor and sends another message to the supervisor stating that it is ready to regulate traffic. This specific position (see Fig. 7.2-(b)) is chosen to allow transporting robots to enter the corridor while at the same time being close enough to the regulator to receive the messages that it emits.

When the regulators in both rooms are ready, the regulation process begins. The supervisor sends messages to both regulators and instructs one of them to let transporting robots enter the corridor, while the other regulator is instructed to prevent robots from entering the corridor from its side. After a certain amount of time, the supervisor sends messages to both regulators to stop robots trying to enter from either side. This allows the corridor to clear before robots from the opposite direction are let through. After another fixed period of time, the supervisor sends a message to the regulators instructing them to allow traffic in the opposite direction of that from the last cycle. After a given number of these switches, both regulators abandon the role and the system goes back to the initial state. This is done so that all robots have a chance to accomplish the task and no robot has the role for the entire duration of the experiment.

In the current version of the implementation, the traffic regulator role allocation procedure relies on the supervisor. Nevertheless, other solutions for a fully decentralized implementation can be found. The simplest of which would be one of the robots with a conflict in the corridor (for instance, always the robot coming from the left room) becoming the supervisor and overseeing the allocation process. By broadcasting a message checking if no other robot had done this before, we could avoid conflicts between multiple robots trying to become the supervisor.

7.3 Self-Organized Approach

We implemented a different solution to our task which does not use institutional roles to regulate traffic. This solution is based on the principles of swarm robotics and the robots rely exclusively on self-organization to solve the task. Conflicts between robots moving in opposite directions in the corridor are solved in the following way: whenever a robot moving in one direction encounters a robot moving in the opposite direction in the corridor, it waits for a period of time proportional to the time that it has been in the corridor. If, during this period, a waiting robot detects that the other robot gives up, turns around and moves back to the room it came from, the waiting robot continues to traverse the corridor. Otherwise, if the time proportional to the time the waiting robot has been in the corridor expires, the waiting robot turns around and heads back to the side of the arena from where it came. No further optimization of the self-organized approach was carried out. For instance, solutions using basic local communication could lead to better performances.

7.4 Controller Design

As previously mentioned, experiments were performed using a FSA controller and only subsequently the IAC was obtained.

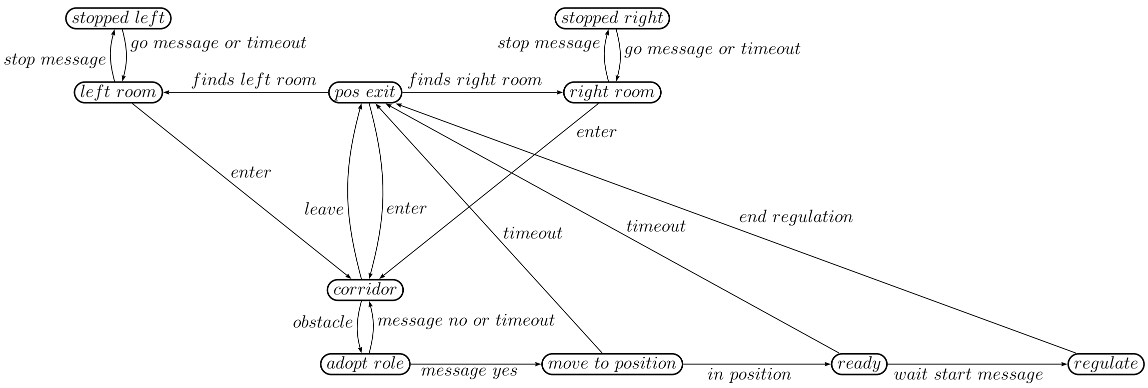


Figure 7.3: FSA controller for the institutional approach.

7.4.1 Finite State Automata

In Fig. 7.3 we display the FSA controller for the institutional approach. The state *pos-exit* corresponds to the state immediately after the robot leaves the corridor, where the robot uses the camera to locate itself inside the arena. We also consider this as the initial state. The FSA represents the behavior described above for navigating in the arena and adopting the role of regulator. There is no direct distinction between states dealing with transporting robots behavior, or those dealing with the adoption and execution of the traffic regulator role. The same is valid for states were the robot is stopped in a queue waiting for a "go" message. Nevertheless, these behaviors clearly have a social interaction component.

In Fig. 7.4 we desplay the FSA controller for the self-organized approach. As in Fig. 7.3, the state *pos-exit* is the initial state.

7.4.2 Institutional Agent Controller

As before, using the IAC methodology to design robotic controllers our aim is to specify behaviors that have a social nature as institutions and behaviors that have an individual nature as the robots' individual behavior.

The individual behavior of the robots specifies how the task at hand is accomplished. Picking up virtual payloads and deploying them is a behavior that has an individual nature, since it relates the robot only to the environment in which it is located. A single robot could accomplish the deployment task, although performance would be critically reduced. Thus, our specification for the individual behavior Ind of the robots is an EPN, displayed in Fig. 7.5-(a), that accomplishes exactly that behavior.

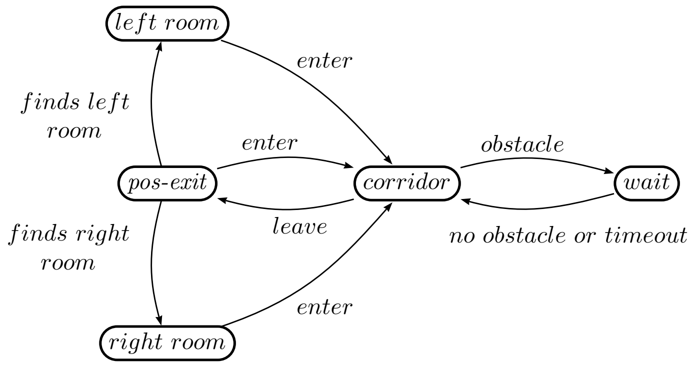
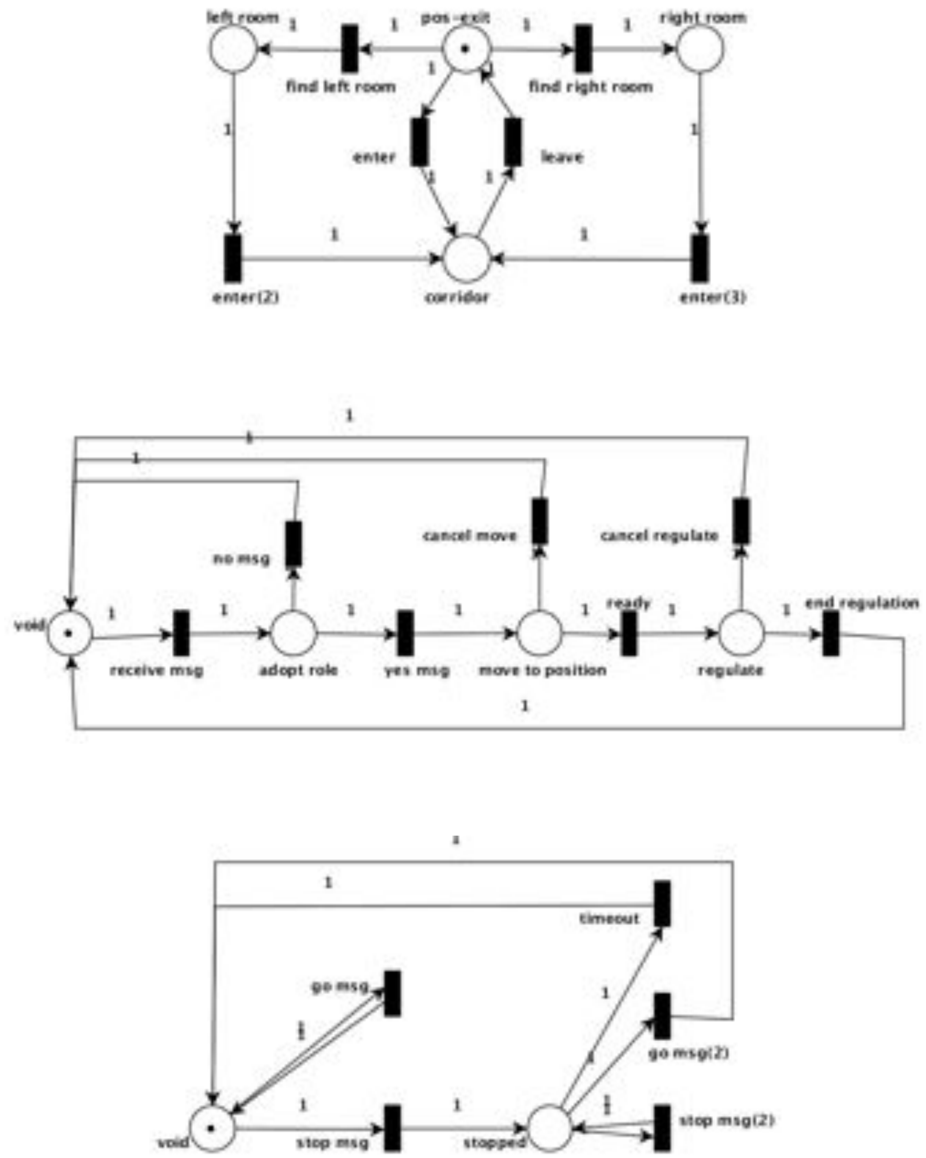


Figure 7.4: FSA controller for the self-organized approach.

The main social behavior of the corridor case study is the traffic regulator institutional role. This is clearly a behavior that has a social nature. We consider that this behavior is specified 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*. The EPN $Inst_R$ to be executed by the robots is displayed in Fig. 7.5-(b). It follows the same sequence of actions described in the previous section and also specified by the FSA controller in Fig. 7.3.

However, the institutional role is not the only social behavior present. As mentioned previously, institutional roles depend on other robots' behaviors, in the sense that other must recognize and/or permit such role playing by particular robots. A second social behavior present in the institutional approach to this task is the recognition and compliance with the traffic regulator. The behavior corresponds to an institution I_M that manages the reception of messages from the traffic regulators and their relay. 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*. The EPN $Inst_M$ to be executed by the robots is displayed in Fig. 7.5-(c).

In Fig. 7.6 we show the higher layer composition of our two institutions and individual behavior. The IAC for this case study is the result of merging this net with the lower layer EPNs. As a feasibility test, we implemented the IAC approach to the case study, having produced similar results to those previously obtained with the FSA approach.



(a)

(b)

(c)

Figure 7.5: Lower layer EPNs for: (a) individual behavior Ind; (b) institution I_R, traffic regulator behavior; (c) and institution I_M, compliance with traffic regulator behavior.

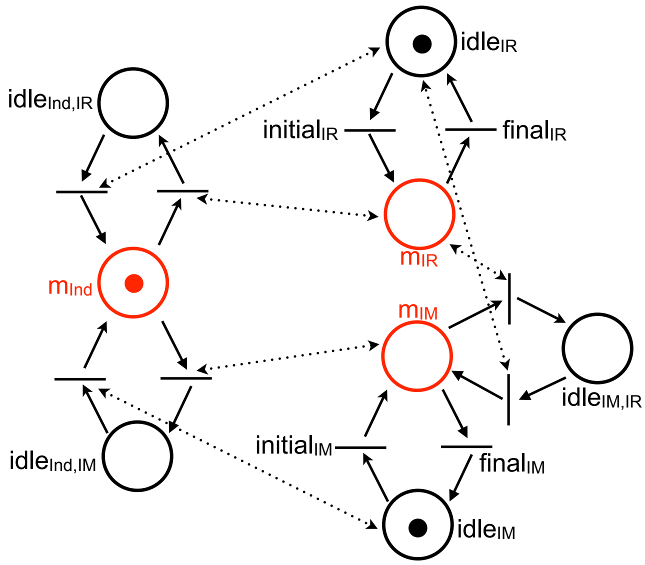


Figure 7.6: Higher layer composition net of IAC for corridor case study. As before, dotted arcs represent bidirectional arcs. Places in red are macro places for behaviors in the lower layer. Place m_{Ind} represents the individual behavior Ind. Place m_{IR} represents institution I_R. Place m_{IM} represents institution I_M.

7.5 Institutional Roles

Experimenting with concepts originated in human institutions in robotic systems has some difficulties. One is to accurately capture the meaning of such concepts, so they can be implemented in robots. It may be difficult in many cases to translate institutional concepts into a robotic implementation, but such an exercise can also help to refine concepts that were somehow vague at its roots in other disciplines. In our example, this translation exercise was needed for the concept of "institutional role". We take an institutional role as a behavior specific to a subset of all robots, that can be seen (by an internal or an external observer) as functional to some collective task or activity, and that depends on other robots' behaviors (in the sense that others must recognize and/or permit such role playing by particular robots).

The corridor case study addresses three crucial issues for institutional roles: role allocation (how some robots start playing a role), role recognition (how robots recognize that some others are playing a role), and role permission (how robots permit other robots to play a role and behave accordingly). These three issues specify one of three elements involved in the ontology of institutional reality (according to Searle, described in Section 3.6): the assignment of status functions. The two others elements are collective intentionality and deontic powers.

Analyzing if robots can display collective intentionality and recognize deontic powers in the same way humans do is a difficult task for the field of artificial intelligence, one that is outside the scope of this work. Nevertheless, such topics must be considered in the next steps of institutional robotics. In this study, we accept that we have some small degree of collective intentionality in the way we program our robots to achieve their goal using the traffic regulator institution. Deontic powers are represented in the definition of institutions and actively enforce prohibitions and permissions. Such representations are a first step in a more comprehensive use of institutions in robots.

In the previous sections we addressed the allocation, recognition, and permission, of the traffic regulator role. We saw that the allocation is made when a conflict occurs inside the corridor, and depends on whether other robots are already executing the role. The recognition of this role by transporting robots is made through a second institution, that associates the reception of a "stop" message with the knowledge that a regulator is in place and is coordinating the team. Role permission is also implied in this second institution, since it specifies how to behave accordingly. Every robot *can* play the role, so in fact every robot *has permission* to do so, meaning that transporting robots will accept the order of the regulator and conform to it.

Another relevant aspect of this case study concerns the distinction between "role" and "individual." A practical difficulty in mounting the experience led to the process we call "role propagation" (described in Section 7.2.3). This process is directly linked to the distinction between role and individual in an institutional environment. The role can be propagated through individuals inside the corridor. Also, regulators abandon the role after some time. This means that no robot is specifically designed to play the role. Playing a role is something justified because of a collective need, not as a right or an inherent feature of any individual.

Within an institutional framework, roles must be distinguished from particular individual robots. Robots are heterogeneous with respect to some features, but fully interchangeable with respect to some other (basic) features. This makes any robot in principle able to play any role, even if some learning can be required to attain full mastery. Robots are redundant in relation to roles. To this effect, different institutional roles must not be allocated by fixed, once for all, mechanisms (e.g., "genetic" mechanisms) but, instead, by institutional assignment of status functions. If this can be implemented, removing specific individuals from the team does not amount to removing specific roles. On the other hand, the addition of individuals with malevolent roles can be countered by a specific feature of institutional roles: for an individual to play a role, other participants must recognize that role as part on the institutional setting, and accept to behave accordingly. The refusal to accept an individual playing a role (because the role is not part of the institutional setting) can be a mechanism to prevent the intrusion of malevolent roles.

7.6 Experimental Setup

We prepared different setups in order to evaluate how parameters such as the size of the robotic team and the length of the corridor affect the performance. Three different corridor lengths L (50 cm, 100 cm and 200 cm) were considered. For each corridor length, we ran experiments with different numbers of robots N (7, 15 and 20 robots). For each of the nine resulting setups, we performed 30 runs for both the institutional approach and for the self-organized approach. Each run had a duration of $T = 900$ seconds.

We considered different corridor lengths in order to evaluate how L influences the number of conflicts inside the corridor between robots, and how the two approaches perform under different conditions. In order to keep the density of robots inside the arena constant, we let the areas of the rooms depend on the size of the robotic team. We increased the size of the arena proportionally to the number of robots in the experiment. The dimensions of each of the two rooms were 50 by 70 cm for $N = 7$, 100 by 75 cm for $N = 15$, and 100 by 100 cm for $N = 20$.

Although the controllers used are the same for all setups, we changed some parameters between setups for the institutional approach. The external program running on the supervisor node is controlled by two different time intervals. Interval δ_1 refers to the time interval where regulators stop robots from entering the corridor when the traffic is moving in the opposite direction in the corridor. Interval δ_2 refers to the time interval where regulators allow robots to enter the corridor. As the corridor length is increased, it will take longer for the transporting robots to traverse the corridor. For longer corridors, both regulators must therefore prevent robots from entering from either side for a longer period of time before the traffic direction is changed. We consider different values for δ_1 of 15, 21 and 42 seconds, for corridors lengths of $L = 50$, 100, and 200 cm, respectively. As the number of robots increase, we would like to have more robots passing through the corridor, in order to avoid robots being stopped most of the time. Therefore, we must increase δ_2. We consider different values for δ_2 of 30, 60 and 80 seconds, for teams of $N = 7$, 15, and 20 robots, respectively.

Videos of the *Webots* simulations for the institutional approach can be found here (for $N = 7$)[1] and here (for $N = 15$)[2]. A video of the *Webots* simulations for the self-organized approach can be found here (for $N = 20$)[3].

7.7 Metrics

In order to analyze the performance of both approaches to our case study, several performance metrics can be used. Since the goal of the task is for the robots to transport virtual payloads from one room to the other, it is intuitive that the most important metric is the number of successful transportations (a pick up in the left room followed by deployment of the virtual payload in the right room) by the team as a whole. However, we also consider other performance metrics, namely the number of conflicts inside the corridor and the duration of the individual transportations.

Definition 18 *The* number of transportations *is the total number of successful virtual payload deployments (a pick up in the left room followed by deployment of the virtual payload in the right room) achieved by the team during a run of the experiment.*

Definition 19 *The* number of conflicts *is the total number of encounters between robots inside the corridor during a run of the experiment.*

When a robot encounters an obstacle inside the corridor, we count that as a conflict. Thus, when two robots moving in opposite directions meet in the corridor, we count two conflicts – one for each robot. We consider two conflicts and not one in this case to

[1] See `http://www.youtube.com/watch?v=dOMkasp5qEI`.
[2] See `http://www.youtube.com/watch?v=sEvLEpM9R-4`.
[3] See `http://www.youtube.com/watch?v=CTqBzeCsbD4`.

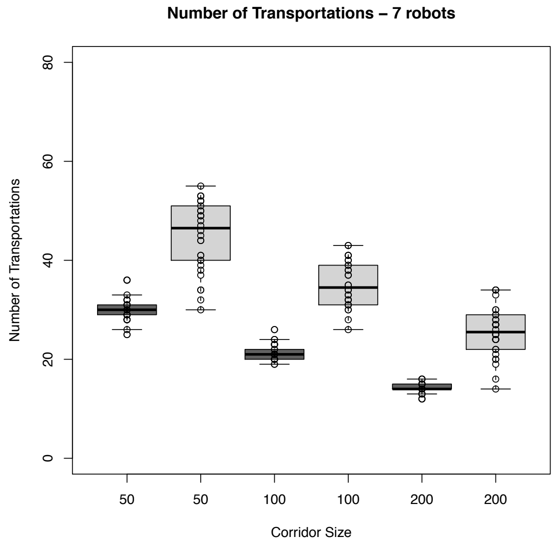


Figure 7.7: Distribution of number of transportations for teams with $N = 7$ robots (institutional in dark grey, self-organized in light grey).

distinguish it from the case in which the obstacle detected by the robot is not located in the corridor but at the entrance.

Definition 20 *The* duration of transportations *is the average time taken by the robots to achieve each individual transportation, from pick up of the virtual payload until deployment.*

7.8 Results and Discussion

7.8.1 Number of Transportations

In Fig. 7.7, we display the number of transportations by a team with $N = 7$ robots in arenas with different corridor lengths. The results for the institutional approach are presented in dark grey while values for the self-organized approach are presented in light grey. The number of transportations decreases as the length of the corridor increases. This is naturally explained by the fact that the robots spend more time traversing the longer corridors.

We observe that the robots following the self-organized approach always manage to perform more transportations than the robots following the institutional approach. This

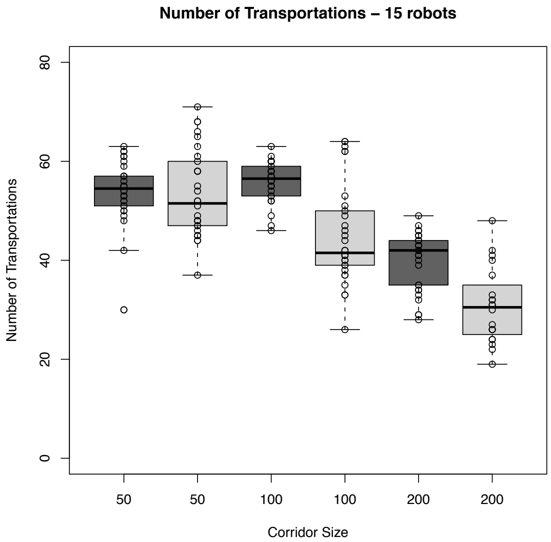


Figure 7.8: Distribution of number of transportations for teams with $N = 15$ robots (institutional in dark grey, self-organized in light grey).

is due to the fact that not all robots in the institutional approach are performing transportations. While in the self-organized approach all the robots are devoted to transporting virtual payload, in the institutional approach two of the robots instead assume the institutional role of traffic regulators. This means that some of the team's resources are spent on coordination. In small teams, a proportionally larger share of robots are dedicated to coordination (28.5% in the case of $N = 7$ robots with 2 traffic regulators). Moreover, in the self-organized approach conflicts are easily solved the first time a (proportionally) large group of robots meets in the corridor, resulting in the robots forming a line and thus making future conflicts rare. This emergent coordination allows the self-organized approach to perform more transportations when the team is small.

In Fig. 7.8, we display the number of transportations performed by a team of $N = 15$ robots. We observe an increase in the number of transportations for both approaches, although the increase for the institutional approach is considerably larger than for the self-organized approach. For $L = 50$ cm, there is not a significant difference between both approaches, except that the variance is greater in the self-organized approach. However, for the longer corridors, the robots following the institutional approach perform more transportations. For teams with $N = 15$ robots, a smaller share of resources are dedicated to coordination (13.3% in this case).

Larger teams have a greater need for regulation than smaller ones, as they are more prone to conflicts occurring often, simply due to their larger number of robots. Since

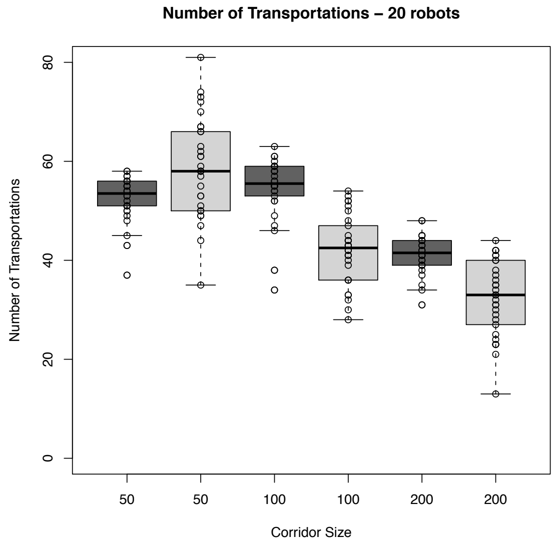


Figure 7.9: Distribution of number of transportations for teams with $N = 20$ robots (institutional in dark grey, self-organized in light grey).

only two robots are devoted to the regulation at any time, larger teams spend less of their resources in coordination than smaller ones. Thus, larger teams have their need for regulation satisfied while allowing a larger share of robots to perform the transport task. The coordination of the team provided by the traffic regulators gives some advantage over the self-organized approach.

The larger variation, with respect to experiments with $N = 7$, in results for the institutional approach is due to the fact that, with a higher number of robots more conflicts occur in the corridor. It is less likely that the first two robots that encounter in the corridor eventually become regulators. Robots may switch the role between them multiple times, leading to a difference in the time that it takes before the traffic regulators effectively start coordinating the rest of the team (and therefore a difference in number of transportations).

Fig. 7.9 displays the distribution of the number of successful transportations for teams with $N = 20$ robots. There is no significant increase in number of transportations for either of the approaches when compared with experiments for teams of $N = 15$ robots, possibly indicating that a stationary state of the system has been reached. Therefore, the advantage of the institutional approach is also maintained for larger numbers of robots. We believe that this is due to rooms' areas becoming too large and robots spending too much time navigating through them. It must be noted that this increase in area is done in order to maintain the density of robots constant.

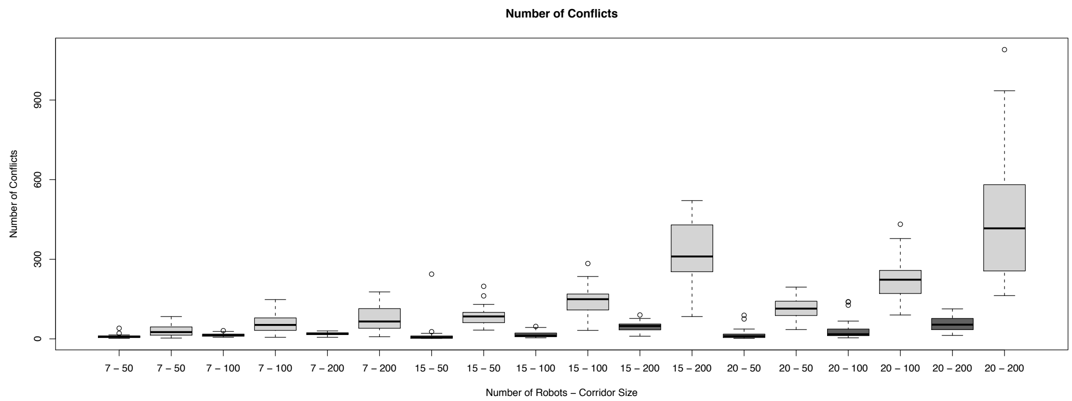


Figure 7.10: Distribution of number of conflicts for different team sizes (institutional in dark grey, self-organized in light grey).

For different sizes of the team and different corridor lengths, we observe that the variance of results is always smaller in the institutional approach than in the self-organized approach. This suggests that the regulation not only positively affects the performance of the system, but also its stability.

7.8.2 Number of Conflicts

In Fig. 7.10, we display the distributions of the number of conflicts for all nine experimental setups. The first 6 boxes from the left display results for teams of $N = 7$ robots, the middle 6 boxes display results for teams of $N = 15$ robots, and the final 6 boxes display results for teams of $N = 20$ robots. The number of conflicts increases with both the size of the team and with the corridor length. In all cases, teams following the self-organized approach have more conflicts than teams following the institutional approach. Since the self-organized approach has no explicit way of controlling the number of conflicts, we can assume that, if we considered the institutional approach with no traffic regulators, the number of conflicts would be at least in the same order of those for the self-organized approach. As the number of conflicts is effectively lower for the institutional approach, we conclude that the coordination effort by the traffic regulators is successfully reducing the number of conflicts.

7.8.3 Duration of Transportations

In Fig. 7.11, we display the average time durations of transportations by teams of $N = 7$ robots for different corridor lengths. The institutional approach spends on average more time in each transportation for small teams than the self-organized approach. This is due to the fact that in the institutional approach robots often have to wait for a message from the regulator to proceed. Moreover, with a small number of conflicts in the self-organized approach (with $N = 7$), robots do not usually stop inside the corridor due to a conflict. The difference between approaches becomes smaller for larger corridor lengths.

Fig. 7.12 displays the average time duration of transportation for teams of $N = 15$ robots. For both approaches, the transportation time increases compared to teams of $N = 7$ robots. This is due not only to the fact that the rooms of the arena are larger, but also that more robots usually queue up before entering the corridor, for teams following the institutional approach. In the self-organized approach, the high number of conflicts inside the corridor prevents the robots from transporting the virtual payload faster. For teams of $N = 20$ robots, both approaches perform similarly when$L = 50$ cm. For longer corridors, the increased number of conflicts results in longer transportation times for the self-organized approach (see Fig. 7.13). For team of $N = 15$ and $N = 20$ robots, the difference between approaches increases with the size of the team.

As in the number of transportations, we can see a difference in transportation times for different team sizes. For smaller teams, few conflicts arise when following the self-organized approach. This allows robots to keep moving constantly and thus reduces the transportation time. However, for teams with more robots, the benefits of coordination become clear. Since in the self-organized approach there is a larger number of conflicts, the robots spend a significant amount of time stopped in the corridor and thereby increasing the transportation time. On the other hand, the coordination in the institutional approach allows most of the robots to keep moving at a steady rate thereby reducing the transportation time.

7.9 Collective Decision-Making with Institutions

In Chapter 4 we argue that using a formal abstracted representation for our definition of institutions, together with a modular approach, allows for changes to institutions to be propagated within a team during execution. Such changes can come about via designer intervention or by collective decision-making within the team, and can be as simple as updating the value of a parameter or as complex as changing the structure of the EPN of a given institution.

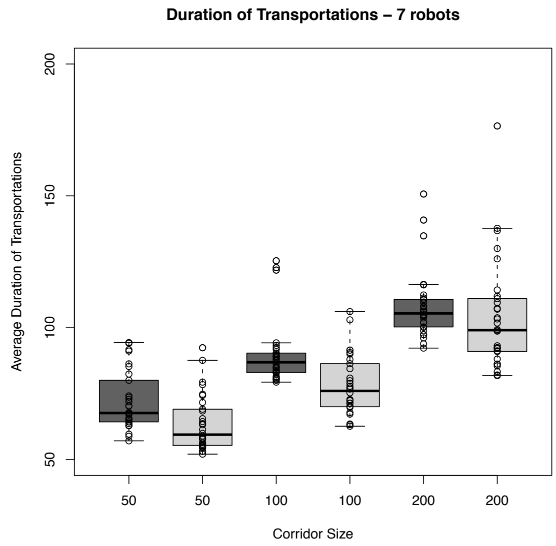


Figure 7.11: Distribution of average time duration of transportations for teams with $N = 7$ robots (institutional in dark grey, self-organized in light grey).

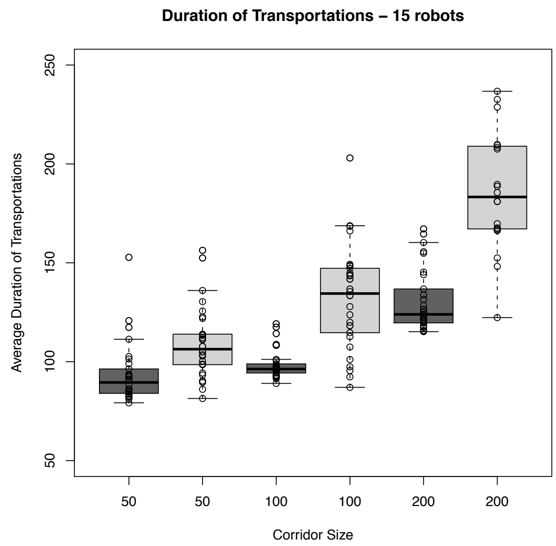


Figure 7.12: Distribution of average time duration of transportations for teams with $N = 15$ robots (institutional in dark grey, self-organized in light grey).

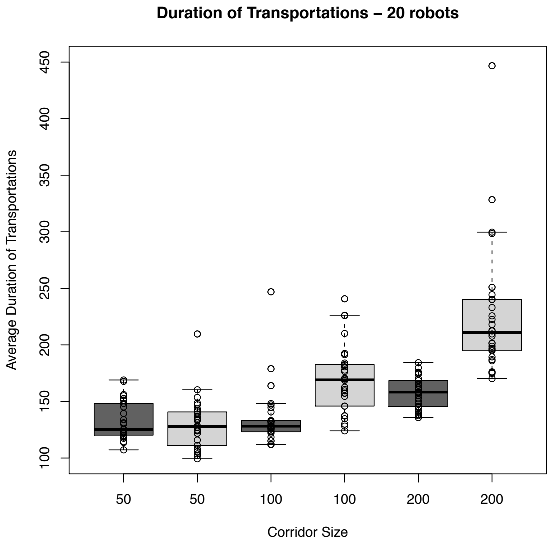


Figure 7.13: Distribution of average time duration of transportations for teams with $N = 20$ robots (institutional in dark grey, self-organized in light grey).

In this section we exemplify the former case. The goal is to allow members of the robot team to cooperatively adapt their behavior to a dynamically changing environment, by updating the value of a single parameter. We consider that the corridor length is modified during execution. This change in the environment has an impact on the coordination performed by the traffic regulators, since it affects the δ_1 time interval that regulates the waiting period for robots at the entrance of the corridor. If this parameter is not adjusted to the proper value of the corridor length, the number of conflicts in the corridor increases or the robots spend an unnecessary amount of time in queues.

To deal with the changes, robots must be able to estimate the length of the corridor. These estimates must be communicated within the team and somehow aggregated to improve the quality of estimation. We use transporting robots to obtain the estimates. Sharing and aggregation of estimates takes place while the robots are in a queue, taking advantage of their temporary static location and proximity to the regulators to facilitate the process. When regulators update the δ_1 parameter the team coordination is modified.

The work described in this section was performed in collaboration with Emmanuel Senft during a semester project at EPFL supervised by José N. Pereira and Prof. Alcherio Martinoli [Senft et al., 2012].

7.9.1 Dynamical Environment & Corridor Length Estimation

We consider the three different values for the corridor length L described in Section 7.6: 50 cm, 100 cm and 200 cm. At a given point during execution, the corridor length is changed in a single time step from a initial length L to a final length L'.

Rather than measuring the corridor length in terms of distance, each transporting robot measures the time it takes to traverse the corridor. The δ_1 parameter can be approximated by this time value, since the time interval during which both regulators simultaneously prevent transporting robots of entering the corridor must be just enough to allow robots to fully traverse it. Initially, robots have a maximum value set for δ_1, designated as δ_M, that would allow no conflicts in the corridor but enforce too much resting time in the queue.

Transporting robots obtain a new estimate of δ_1 each time they traverse the corridor without any conflict. If a conflict occurs in the corridor the robots involved set their estimates to δ_M. Individual estimates must be aggregated so that the regulator changes the regulating periods appropriately.

7.9.2 Communication & Collective Decision

Sharing of estimates between members of the team takes place while the robots are in a queue. At this stage of their behavior, robots already exchange messages, propagating the stop message originally sent by the regulator through the queue. In order to share estimates, robots append to the message their current value of δ_1. These messages are sent using the short range communication, which leads to robots receiving messages from only the two robots immediately in front and behind (in terms of positions in the queue).

The set of robots communicating with robot i at each iteration k of the message exchanging protocol is denoted as $C_i(k)$ and its size as $c = |C_i(k)|$. Robot i stores the estimates of δ_1 shared by its teammates at iteration k in a vector $E(k) = [e_0(k) \; \dots \; e_c(k)]$, where $e_j(k)$ is the estimate shared by robot $j \in C_i(k)$, if $j > 0$, and $e_0(k)$ is the current δ_1 estimate of robot i.

The collective decision on the estimate of δ_1 to be used comes from each robot solving a consensus problem [Ren et al., 2005], with the goal of all robots agreeing on a value for the parameter. This means that, at each iteration k of the message exchanging protocol, each robot updates its estimate $e_0(k)$ of δ_1 in the following manner:

$$e_0(k+1) = \sum_{j=0}^{c} \beta_{ij}(k) e_j(k), \tag{7.1}$$

Table 7.1: Correspondence between corridor length L and time necessary to traverse the corridor δ_1.

L [cm]	δ_1 [s]
50	10
100	20
200	40

where $\beta_{ij}(k)$ is a weighing parameter that we fix as $\beta_{ij}(k) = \frac{1}{c+1}$, for all i, j. This update is performed at every iteration, except when the estimation from another robot is δ_M, indicating that that robot had a conflict inside the corridor. In that case, robot i updates its estimate as $e_0(k+1) = \delta_M$, causing this maximum value to propagate through the queue and indicate to regulators that the δ_1 period must be extended to the maximum in order for robots to fully navigate the corridor and obtain proper estimates of its length.

Regulators are part of the collective decision process, sharing and updating their δ_1 estimates although not navigating through the environment. However, this parameter must be the same in both regulators, since their periods of allowing or preventing robots from entering the corridor must be synchronized. Given that their estimates of δ_1 might be different, we choose to use only the estimate of the regulator in the left room. This robot adds a fixed margin to its estimate of δ_1, allowing more time for robots exiting the corridor to vacate the path taken by robots navigating in the opposite direction. This new value is sent to the supervisor node which enforces the synchronization between regulators.

7.9.3 Results and Discussion

To analyze the adaptation of the team to changes in the environment we tested two different setups. In the first, we consider a small decrease in corridor length from $L = 100$ cm to $L' = 50$ cm at time $t = 600$ seconds. In the second, we consider a big increase in corridor length from $L = 50$ cm to $L' = 200$ cm at time $t = 450$ seconds. Different corridor lengths correspond to different real values of δ_1, indicating the time necessary to traverse the corridor. A correspondence between L and δ_1 is shown in Table 7.1. In both setups we consider $N = 7$ robots and perform 10 runs of the submicroscopic simulation while keeping track of each robot δ_1 estimate.

In Fig. 7.14 we display the evolution of the estimates of δ_1 for the first setup (small decrease in corridor length). The top two plots display results for a single run. The top plot compares the evolution of the average of estimates for all the robots to the real value of δ_1. The middle plot displays the evolution of each individual robot estimate. The bottom plots shows the mean error (averaged over the 10 runs performed) between the

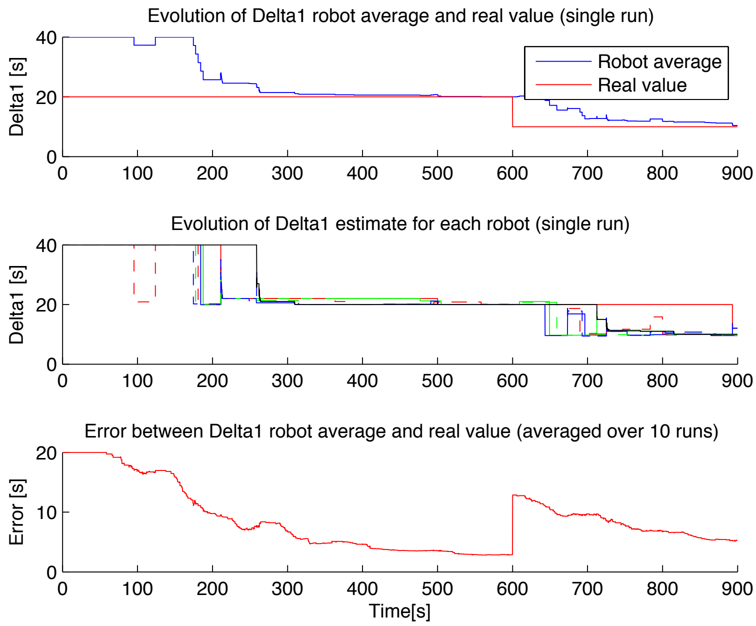


Figure 7.14: Analysis of δ_1 estimates for small decrease in corridor length setup. Top: comparison between average of robot estimates and real value for δ_1. Middle: detailed view of each robot estimate. Bottom: mean error between robot average and real value.

real value and the robot estimate average of δ_1.

All robots are initialized with a δ_M maximum value for δ_1, causing them to have to adapt their estimates to the proper corridor length even before any change in the environment. Adaptation does not start immediately since for robots to obtain proper estimates they must traverse the corridor without conflicts, which most often requires the regulators to be present to coordinate traffic. We can observe that both individual estimates and robot average converge to the real value of δ_1. After the change in corridor length at $t = 600$ seconds the robot team again adapts to the newly set length and corresponding value of δ_1. Although never reaching zero, the mean error over several runs follows this same tendency.

In Fig. 7.15 we display the evolution of the estimates of δ_1 for the second setup (big increase in corridor length). We follow the same data representation scheme used in Fig. 7.14. We can again observe that the team is able to adapt both to the initially set corridor length L as well as the modified length L'. In the middle plot of Fig. 7.15, we observe that, once the change in corridor length happens, some robots quickly update their estimate to the the new value of δ_1, while for others this happens only later and not in a single

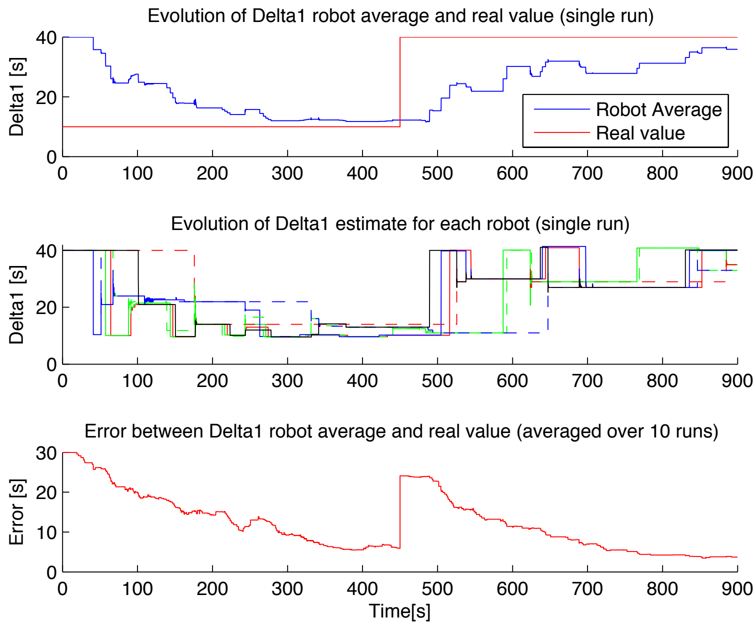


Figure 7.15: Analysis of δ_1 estimates for big increase in corridor length setup. Top: comparison between average of robot estimates and real value for δ_1. Middle: detailed view of each robot estimate. Bottom: mean error between robot average and real value.

update. This is due to the fact that robots that have a conflict in the corridor change their estimate to δ_M.

Summary

In this chapter we have demonstrated how concepts from institutional robotics can be applied in a robotics task, focusing on one specific form of institution, namely the institutional role. We have shown that coordination artifacts set up as institutional roles can effectively help a robotic team organize and improve performance in a given task. Nevertheless, this not true in all cases. For instance, we showed that for smaller teams, emergent coordination from a set of simple control rules is sufficient for the team to achieve a good performance. With the increase of the size of the robotic team, and the consequent decrease in proportion of robots devoted to institutional roles, we see benefits of using institutional roles, not only in the overall performance of the task but also in its stability. In the final section of the chapter we presented a modified version of the case study in which robots adapt their controllers to accommodate for changes in the environment. This study provides a relevant first step towards autonomous adaptation of institutions.

Chapter 8

Considering Institutions in Unsustainable Robotic Systems: the Piece Assembly Case Study

In this chapter we present a case study originally introduced in the scope of the project "From Bio-Inspired to Institutional-Inspired Collective Robotics"[1] (BioInstBots). The aim of this project was to study the laws governing collective systems, with the goal of translating such laws into multi-robot systems that could display complex collective behaviors. To do so, we took inspiration from two distinct research fields: cell populations in biological systems; and institutional economy in social systems.

A key property of cell populations is their sustainability, which is related to the ability of a system to keep its population alive. Different types of cells must be present in certain numbers and ratios so that the overall system is sustainable. Sustainability also appears in social systems. It is usually linked to the ability of a population to maintain its numbers despite diminishing or finite available resources.

The standard class of such social dilemmas is the "tragedy of the commons" [Hardin, 1968], which has been studied extensively under the light of institutional economics, primarily by Elinor Ostrom et al. [Ostrom, 1990, Ostrom, 2005, Dietz et al., 2003, Anderies et al., 2004]. Of particular interest to this case study is the work presented in [Poteete and Ostrom, 2004], where the effects of heterogeneity, group size, and the role of institutions, on the sustainable management of resource-dependent populations is studied.

We envision that sustainability will also be an important property in truly social human-

[1]FCT-sponsored project (Ref: PTDC/EEA-CRO/104658/2008) - http://mediawiki.isr.ist.utl.pt/wiki/From_Bio-Inspired_to_Institutional-Inspired_Collective_Robotics

robot mixed systems, where robots act as independent entities with their own goals. This case study is aimed at studying the sustainability of a robotic system, that sustainability being dependent on the heterogeneity of the robotic team, and the impact of considering institutions in the system.

To do so, we consider a transport and assembly task, that we designate as the *piece assembly case study*. Essentially, robots in a heterogenous team need to collect *components* of different types, which are the basic building blocks needed for *pieces* to be assembled. These are present at specific locations of the environment, and must be transported to a particular area called *assembly site*, where the assembly process takes place. The team goal is to maximize the number of pieces assembled.

Robots spend energy while moving through the environment and are rewarded with energy, both when they deliver components to the assembly site and when pieces are correctly assembled. In some situations, either by virtue of the physical environment or by virtue of choices of individual robots, the energy level of some robots will drop below zero and they will become non-operational.

We consider that the sustainability metric describes the ability of a robot team to keep its members operational during run time. A second metric – efficiency – will relate directly the team performance, measured as the number of pieces assembled, to the total energy spent.

As in the tragedy of the commons, there is a social dilemma of how to explore the resources present in the environment (components). Robots can give priority to their individual goal of remaining operational or to the collective goal of maximizing team performance. The heterogeneity in our robotic team comes from considering two types of robots that take different decisions regarding this dilemma, one type giving priority to their individual goals and a second type giving priority to the collective goal.

We propose two approaches to this task that differ only in the manner in which the piece assembly process occurs.

- A fully *decentralized* approach decomposes the process into individual decisions taken by the robots delivering the components.
- On the other hand, an *institutional* approach puts the burden of the piece assembly on one robot from the team, using an institution designed for this purpose, while the remaining robots on the team merely transport the components to the assembly site.

Since the institutional approach requires an extra effort in terms of coordination, we consider that using the specified institution has an associated cost in terms of energy.

Our objectives for this case study, to be tackled in the subsequent sections, can be summarized as assessing answers to the three following questions:

- How does heterogeneity in a robotic team affect its efficiency and sustainability when performing a given task?
- What impact can the introduction of institutions have on the efficiency and sustainability of a robotic team that is not efficient and not sustainable?
- What costs of these institutions are acceptable without becoming too heavy for the robotic team to be efficient and sustainable?

This case study is studied using spatial microscopic simulations implemented through a very abstracted, point-based simulator. This more abstracted simulator was chosen to accommodate the use of the case study implementation in different studies of the BioInstBots project, some being quite computationally expensive.

In the final sections of this chapter we address two further problems. First, we study how robust our approaches are to changing environments when dealing with complex tasks involving a social dilemma. Second, we observe how our modeling methodology can be applied to obtain a-priori estimates of the performance of the team in different environments.

This chapter is organized as follows. Sections 8.1 and 8.2 describe the environment in which the robots operate, and the task they must accomplish, while Section 8.3 describes the metrics used to assess our objectives. Sections 8.4 and 8.5 present the two approaches to the task, with controller design being discussed in Section 8.6. In Section 8.7 we discuss in more detail how conflicting goals and a heterogenous population affect this task. Experimental setup and results for this case study are presented in Sections 8.8 and 8.9. As mentioned, in Sections 8.10 and 8.11 we discuss robustness and modeling. Section 8.12 ends the chapter with a general discussion on our objectives and results obtained in previous sections. In Table 8.1 we present some useful parameter notation used throughout this chapter.

8.1 Environment & Robots

To perform our microscopic simulations we use a stochastic, spatial (2D), kinematic, discrete-time, multi-agent simulator previously described in Chapter 2. Given the very abstracted nature of the simulator, we choose to use arbitrary units designated as *unit* for distance and *time-step* for time. The simulated environment contains point representations of the assembly site, component wells and robots. The assembly site is the point in the environment where piece assembly is performed. Component "wells" are points where the robots can obtain the necessary components to complete pieces. At each well i

Variable	Description
N	Number of robots
T	Simulation time horizon (in time steps)
C	Number of components
S	Number of slots at assembly site
D	Distance between assembly site and component wells
r_d	Detection radius for assembly site and component wells
e	Energy consumed per time step by one robot
V_m	Maximum velocity of the robots
CR	Collective reward
IR	Immediate reward

Table 8.1: Parameter notation table.

($i = 1, \ldots, C$) robots obtain components of type i available only at that well. We consider wells to have an infinite supply of components.

We consider the assembly site the "center" of the environment and place the component wells in a circle of fixed radius (D) around that center and separated by a fixed angle ($2\pi/C$). The assembly site is divided as a sequence of indexed slots (indexes between 1 and S), each slot being able to accommodate a component. If all the slots are occupied and no piece has been completed the assembly site is cleared. Fig. 8.1 displays an initial snapshot of the microscopic simulation environment.

Robots are initially scattered randomly throughout the environment. We make several assumptions about robots' behavior and operation.

- There are two types of robots, which give different priorities to individual or collective goal, as we will see further ahead. We designate them as *short-sighted* and *far-sighted* robots.
- Robots move in a holonomic fashion. We consider an obstacle avoidance behavior, active at all times in the robots, which considers robots has having a circular shape with a 0.375 *unit* radius.
- The energy consumption feature of our simulated robots is considered in a highly abstracted manner. Our robots consume one unit e of energy at every time step of the simulation, irrespectively of what behavior they are executing. The value of this unit e is related to the energy rewards the robots obtain at the assembly site.
- We assume that all robots know the location of component wells and assembly site.

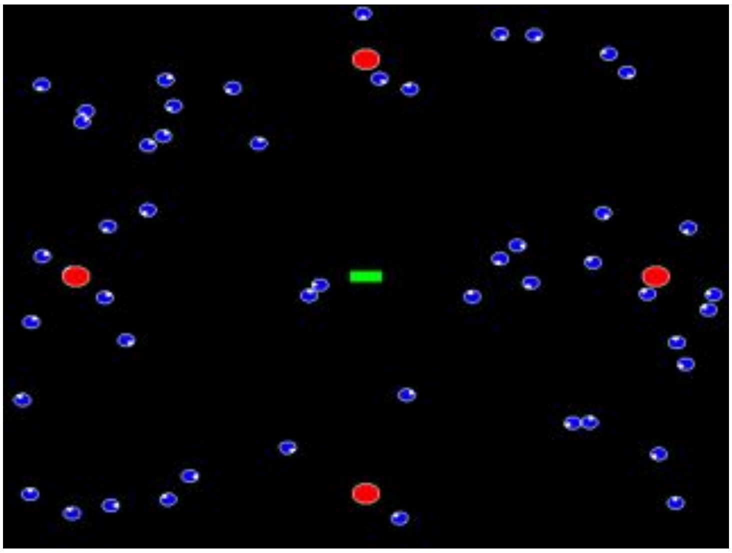


Figure 8.1: Snapshot of microscopic simulation environment with assembly site (green rectangle), component wells (large red circles) and robots (small blue circles). The environment is toroidal, wrapping both on the vertical and horizontal edges.

Note that this last assumption does not interfere with the purpose of our study. We do not wish to reach any conclusions about the way that different approaches (centralized vs decentralized, global information vs local information, etc) perform target identification and retrieval tasks, or foraging tasks, or coverage tasks. Since the assumption about knowledge of component wells and assembly site is valid for both approaches we will be tackling (decentralized and institutional), we assume it does not have an impact in our study.

Energy rewards are given to the robots both when they deliver components to the assembly site and when a piece is correctly assembled. We make a clear distinction between these two types of reward: rewards obtained when delivering components to the assembly site are called *immediate* rewards; rewards obtained upon piece completion are called *collective* rewards. Collective rewards (CR) are given when the last component of a piece is placed and are divided by all the robots that took part in the completion of that piece (e.g., if three robots contributed to a certain piece being completed they would be given each an energy reward of $CR/3$). On the other hand, immediate rewards (IR) are given immediately after a robot has delivered a component to the assembly site. The value of this reward is equal to the index of the slot where the component is placed by the robot (slot i gives reward i). Different values for both rewards will lead to different decisions from the robots. Moreover, the most relevant difference between both types of rewards is the temporal aspect. While robots are immediately rewarded when they place a component in the assembly site (with larger rewards depending on where they place them), the collective reward coming from completing a piece is only given in the future (and might not be given at all). The robots' decisions must take this into account.

As stated above, the value of the unit e of energy consumed per time step by one robot is related to the immediate energy rewards obtained at the assembly site. We consider that the maximum possible immediate reward (equal to the index of the last slot S) should give the robot sufficient energy for a new round trip between the assembly site and a component well. This means that $e\,T_v = S$, where T_v is the number of time steps necessary to perform a round trip visit between assembly site and a component well. T_v is calculated as $T_v = 2D/V_m$, allowing us to calculate a value for e. Obviously, T_v does not represent the actual time taken for a robot to complete this trip but rather a lower bound. It is also clear that, most times, robots will not be able to place their component on the maximum immediate reward slot and will not obtain sufficient energy for a new trip. To prevent the robots from becoming immediately non-operational when this happens we endow them with an initial reserve of energy. This initial reserve is calculated to allow the robots to perform 25 round trips between assembly site and a component well. Nevertheless, this reserve will also run out and so the robots must perform the task to obtain the needed energy rewards.

8.2 Task Description

The piece assembly task carried out by the robots can be seen as having two distinct stages.

The first stage consists of a transportation task. Robots have to locate and reach the component wells in order to obtain the necessary components for piece assembly. After reaching the well and collecting a component, robots have to reach the assembly site in order to deliver the component. Since our simulated robots have no manipulation capabilities the collection, transport, and delivery, of components happens only in a "virtual" way. When robots are inside a certain detection radius of the wells (r_d), they collect a "virtual" component. When they are inside the same radius of the assembly site, they deliver that same component.

The second stage of the task is the placement of a collected component in the assembly site in order to assemble pieces and obtain energy rewards. It is at this stage that cooperation between robots becomes necessary in order to successfully achieve the overall system goal of assembling pieces. A piece is considered successfully completed when sequential slots of the assembly site have all the existing types of components organized in descending order, as presented in Fig. 8.2. Consider that slot i has a component of type 1. A

component type		$j=C$	$j=C-1$		$j=2$	$j=1$	
occupied slot	$\cdots$	C	$C-1$	$\cdots$	2	1	$\cdots$
slot number		$i-C+1$	$i-C+2$		$i-1$	i	

Figure 8.2: Abstracted piece completed representation in assembly site. Initial slot indicated by i with component of type 1. Component type varies with j, $j = 1, \ldots, C$.

piece is complete if all slots $(i - j + 1)$ $(j = 1, \ldots, C)$ have a component of type j, with the condition $i \geq C$ being verified (indicating that before the initial slot i are enough slots to accommodate a complete piece). Slots with higher index and higher immediate reward will accommodate components with lower index. When this happens, the number of pieces completed is increased, the slots used for the piece are cleared, and each robot receives an energy reward of CR/C for each component delivered that was used for that particular piece.

8.3 Metrics

The performance of the robot team in the task can be directly measured by the number of pieces correctly assembled divided by the simulation time horizon. We call this metric the rate of piece completion. Nevertheless, when designing multi-robot systems, performance must be considered in view of parameters such as energy cost, complexity, robustness, and others. Apart from rate of piece completion, we are interested in efficiency and sustainability.

8.3.1 Efficiency

Within our highly abstracted implementation we can consider the energy costs of our system by adding the energy spent by all the robots (E_R) and measure the energetic efficiency of the system. To do so, we consider a metric designated as efficiency.

Definition 21 *The* efficiency *metric ($\mathcal{E}$) relates the number of pieces completed (P) with the energy spent in the following manner:*

$$\mathcal{E} = \frac{P}{E_R} \tag{8.1}$$

Since the units of P and E_R are not directly related, making this ratio not ideal in terms of comparison of different approaches, we calculated an upper bound for the efficiency metric. This will allow us to compare different approaches using a normalized version of efficiency.

In order to obtain this upper bound we make two assumptions about the "ideal" conditions for high efficiency. First, we assume that robots would travel directly in a straight line between the assembly site and the component wells. Second, we assume that components would arrive at the assembly site in the ideal order for piece completion - first a component of type 1 would arrive, followed by a component of type 2, and so forth, until a component of type C arrived and a piece was completed. The next component to arrive would be again of type 1 and this would repeat during run time. Following these assumptions, we would have the maximum possible number of pieces completed, since the number of visits to the assembly site needed to complete a piece would be minimal (C, one for each necessary component) and the time between visits also minimal. Note that not only the number of pieces completed would be maximal but also the energy spent by the robots (since all robots are operational all the time). One could argue that an upper bound for efficiency should consider a lower value for E_R but that would indicate that some robots became non-operational during simulation time which would also cause the number of completed pieces to decrease.

To calculate the maximum efficiency ($\mathcal{E}_{max}$) we compute the maximum number of pieces completed (P_{max}) and the maximum energy spent (E_{max}). The latter is easily obtained since we consider that robots are operational for the duration of the simulation ($O_i = T$, where O_i is the number of time steps robot i is operational). Considering the value for e we have:

$$E_{max} = e\,T\,N = \frac{S\,T\,N}{T_v} \tag{8.2}$$

In order to calculate P_{max} we only need to know the maximum number of visits (Vis_{max}) that can be made to the assembly site. Given that, and following our assumption, at each C visits a new piece will be completed, and P_{max} can be calculated as $P_{max} = \text{Vis}_{max}/C$. Vis_{max} can be obtained from the simulation time horizon and the time needed to complete a visit as is shown next.

$$P_{max} = \frac{\text{Vis}_{max}}{C} = \frac{\frac{T\,N}{T_v}}{C} = \frac{T\,N}{T_v\,C} \tag{8.3}$$

Combining (8.1),(8.2), and (8.3) we can calculate $\mathcal{E}_{max}$:

$$\mathcal{E}_{max} = \frac{P_{max}}{E_{max}} = \frac{\frac{T\,N}{T_v\,C}}{\frac{S\,T\,N}{T_v}} = \frac{1}{C\,S} \tag{8.4}$$

We conclude that, in our setup, the upper bound for task efficiency depends only on the number of components and number of slots of the assembly site. These parameters are linked directly to the complexity of the assembly process - the less components and slots, the simpler the task. In the extreme case of only one type of component and one slot on the assembly site, the $\mathcal{E}_{max}$ is 1. Using $\mathcal{E}_{max}$ we can obtain a normalized value of efficiency for each simulation performed. Using P_{max} we can also obtain a normalized value for the rate of piece completion.

8.3.2 Sustainability

In our case study, we consider that the sustainability of our system describes the ability of the robot team to keep its members operational during run time.

Definition 22 *The* sustainability *metric ($\mathcal{S}$) is the mean number of time steps agents are operational (averaged over the number of robots N), calculated as follows:*

$$\mathcal{S} = \frac{1}{N}\sum_{i=1}^{N} O_i \tag{8.5}$$

The upper bound for sustainability ($\mathcal{S}_{max}$) is T, obtained when all robots remain operational throughout a simulation. Using this upper bound we obtain a normalized value of sustainability for each simulation performed.

8.4 Decentralized Approach

The decentralized approach decomposes the assembly process into individual decisions taken by the robots delivering the components. Another way to say this is that the piece assembly process is distributed by the robot team, allowing it to proceed even if any (or the majority) of the robots becomes non-operational.

In the first stage the robots have to choose a component well and move towards it. After reaching a component well, robots collect a component and move to deliver it to the assembly site. This choice of a well is performed randomly and according to the distance of the wells to the assembly site. If all the wells have the same distance to the assembly site (as we have specified for our environment), then the probability of a robot choosing any particular well is $1/C$. Nevertheless, further ahead we will observe what is the impact of having wells at different distances to the assembly site, so a more general way of computing the probabilities of choosing each well is needed.

Our intuition is that the probability of a robot choosing a particular well should decrease as the distance of the well to the assembly site increases. This represents that detection of a component well using onboard sensors should be dependent on distance. Consider d_i the distance of component well i to the assembly site and the auxiliary parameter $d' = max(d) + min(d)$, where $d = (d_1, \ldots, d_C)$, and $max(d)$ and $min(d)$ represent the maximum and minimum distance of the assembly to a well ($max(d) = min(d)$ if all wells are at the same distance), respectively. The probability p_i of a robot choosing component well i is given by:

$$p_i = \frac{d' - d_i}{\sum_{i=1}^{C}(d' - d_i)} \tag{8.6}$$

In the second stage, robots have to place the component they have collected in one of the assembly site vacant slots. Until this point, robots execute the task in a reactive manner, but now a choice has to be made that depends on their own goals. Each robot can give priority to the collective goal of completing pieces, possibly receiving a low immediate reward, or give priority to its individual goal, preferring to obtain a higher immediate reward and more easily assuring its continued operation, although possibly damaging the efforts of the robot team. This decision reflects a conflict of interests between individual robot and the team.

The state of the placement problem is defined as $(F, C_1, \ldots, C_S)$, where $F \in \{1, \ldots, C\}$ is the component the robot will place at the assembly site, and $C_i \in \{\phi, 1, \ldots, C\}$ is the current component in slot i of the assembly site, ϕ representing a vacant slot. The set of actions available to the agent can be defined as $\{a_{F,1}, \ldots, a_{F,S}\}$, $a_{F,i}$ representing placing component F in slot i. For each action $a_{F,i}$, $i = 1, \ldots, S$, the robot computes a possible reward value r_i it may receive if it places the collected component in slot i in the following manner:

$$r_i = \begin{cases} IR_i + P_i * \gamma * \frac{CR}{C} & \text{, if } C_i = \phi \\ 0 & \text{, otherwise} \end{cases} \tag{8.7}$$

If the slot is occupied the reward value is 0, so we focus on calculating rewards for vacant slots. The first term of the sum (IR_i) corresponds to the immediate reward the robot obtains for placing component F in slot i. The second term of the sum ($P_i * \gamma * (CR/C)$) corresponds to a possible future reward the robot will obtain if a piece is completed using the component it is placing at the assembly site. Since this reward will be obtained only in the future, the collective reward (divided by the number of components) CR/C is weighted by a *discount factor* $\gamma \in [0, 1]$. This factor represents the importance the robots will give to latter rewards, when placing a component in a given vacant slot. In Section 8.7 we discuss how the choice of this parameter impacts different decisions made by the two types of robots (short-sighted and far-sighted). P_i represents the prospect that the collected component F placed in slot i contributes to a piece being completed given the current state of the assembly site, with $P_i = 1$ if this is true and $P_i = 0$ otherwise. Algorithm 3, presented in Appendix B, computes the prospect vector for all slots of the assembly site.

Thus, if placing the collected component F in a slot i allows a piece to be completed, r_i will have contributions both from immediate and collective rewards. Otherwise, the only contribution to r_i will come from the immediate reward. To decide in which slot to place the component, the robot computes r_i, $i = 1, \ldots, S$, and chooses the slot with maximum possible reward.

The use of Algorithm 3 to decide if placing a component in a given slot contributes to the completion of a piece can be seen as enforcing a policy defined a priori for piece assembly. Although some experiments were performed with active learning of a policy for piece assembly, we have chosen not to focus this work on the subject of decentralized learning and/or learning of institutions. Nevertheless, we observed that, given a longer simulation time horizon, results obtained with active learning converged to those obtained with the a priori defined policy. It must also be noted that, using our a priori defined policy, we consider every delivery of a collected component by a robot to the assembly site as a independent event. We do not consider sequences of states, actions, and rewards in our approach. At any delivery, robots make a decision based only on their collected component and the current state of the assembly site.

8.5 Institutional Approach

The institutional approach puts the burden of piece assembly on one robot from the team, using an institution designed for this purpose, which we will designate as *institutional assembler*. Other robots merely transport the components to the assembly site, where the robot executing the institution conducts the assembly process.

The *institutional assembler* is an institutional role performed by one of the members of the robot team. We randomly choose a robot to perform the role of institutional assembler (effectively becoming *the* institutional assembler) for the duration of each simulation. The institutional assembler role is general, in the sense that any robot in the team can perform the role and no specialized robot is needed. For instance, if the robot performing the role malfunctioned and stopped carrying out its duties, any other robot would be able to take its place as institutional assembler. This generality ensures us that the institutional approach is able to deal with some problems of centralized approaches, since the robot performing the role is not a critical point of failure of the entire system.

As in the decentralized approach, the piece assembly task is divided in two stages: transport and assembly. The first stage of the task is handled very much in the same manner as in the decentralized approach. Robots travel between component wells and assembly site, collecting components and delivering them. The difference is that, instead of placing the collected components on the vacant slots of the assembly site, robots deliver the components to the institutional assembler.

The institutional assembler role is performed in the following manner. Rather than traveling through the environment collecting components, the institutional assembler remains at the assembly site waiting for components collected by other robots to be delivered. Its goal is to complete pieces, using these components. To do so, the institutional assembler divides the assembly site in groups of C sequential slots, in order to assemble $\lfloor S/C \rfloor$ pieces concurrently (number of slots S might not be divisible by number of components C). The first group is comprised of slots $1, \ldots, C$, the second group comprised of slots $C+1, \ldots, C+C$, and so on. Each slot $i+j$ ($i = 0 : C : C * [\lfloor S/C \rfloor$-1], $j = 1, \ldots, C$, where $a : b : c$ represents constant increments of size b starting from a until c is reached) is reserved for a component of type j, ensuring that, when a group of sequential slots is complete, a new piece is fully assembled. When a new component of type j is delivered at the assembly site, the institutional assembler places it in one the reserved slots for type j components.

If a delivered component is not needed at that moment (slots for type j components all occupied), the institutional assembler holds the robot delivering that component in a queue, so that the component may be used in the future. Robots do not remain in the queue for an indefinite amount of time, after a certain number of time steps they are released. This number of time steps is randomly drawn from an exponential distribution (with rate $1/T_v$) every time a robot is put in the queue. Once a piece is completed the institutional assembler clears the corresponding slots and searches the queue for available components. If such components are found, they are placed in the appropriate slots and the robots delivering those components are released from the queue.

As in the decentralized approach, immediate rewards are given as soon as a component is placed on the assembly site. Collective rewards are given upon piece completion. However, in the institutional approach, the collective reward is not fully divided by the robots that have delivered the used components. Since the institutional assembler is also a member of the robot team, it also needs to obtain some reward. To do so, the institutional assembler reserves a percentage of the collective reward for itself. We designate this value as the

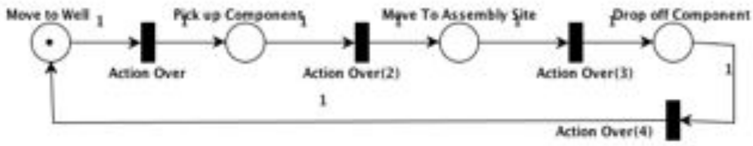


Figure 8.3: EPN controller executed by robots following the decentralized approach.

assembler fee (AF). The remainder of the collective reward is then distributed by the appropriate robots. The assembler fee represents the institutional assembler's "struggle for survival" as part of the robot team. Moreover, and more important, it represents that institutions are not free of costs, that for improved regulation and coordination, robots might have to abdicate of a certain part of their rewards.

8.6 Controller Design

In Fig. 8.3 we display the EPN controller executed by the robots following the decentralized approach. It follows a sequential series of actions - move to well, pick up component, move to assembly site, drop off component - already described in Section 8.4. The first three actions address the first stage of the task, collecting and transporting components to the assembly site. The action "drop off component" encapsulates the decision about placement of the collected component in one of the vacant slots at the assembly site, addressing the second stage of the task. A more detailed controller could have been used for this stage but would not provide any additional insights to our case study. Note that for this approach we use only an EPN controller, not an IAC.

In the institutional approach we consider one individual behavior and one institution – the institutional assembler role. However, and since role assignment has already been discussed in the previous case study, we consider that one single fixed robot executes the institutional assembler institution, while all other robots execute only the individual behavior. We consider an EPN specifying the individual behavior and an EPN specifying the institution. We skip the remaining definition of institution (initial and final conditions, deontic operator) and the composition of individual behavior and institution into an IAC, procedures already described in Chapter 4 and Sections 6.2 and 7.4.

Transporting robots execute only the individual behavior, also specified by the EPN displayed in Fig. 8.3, but with the "drop off component" action being substituted by

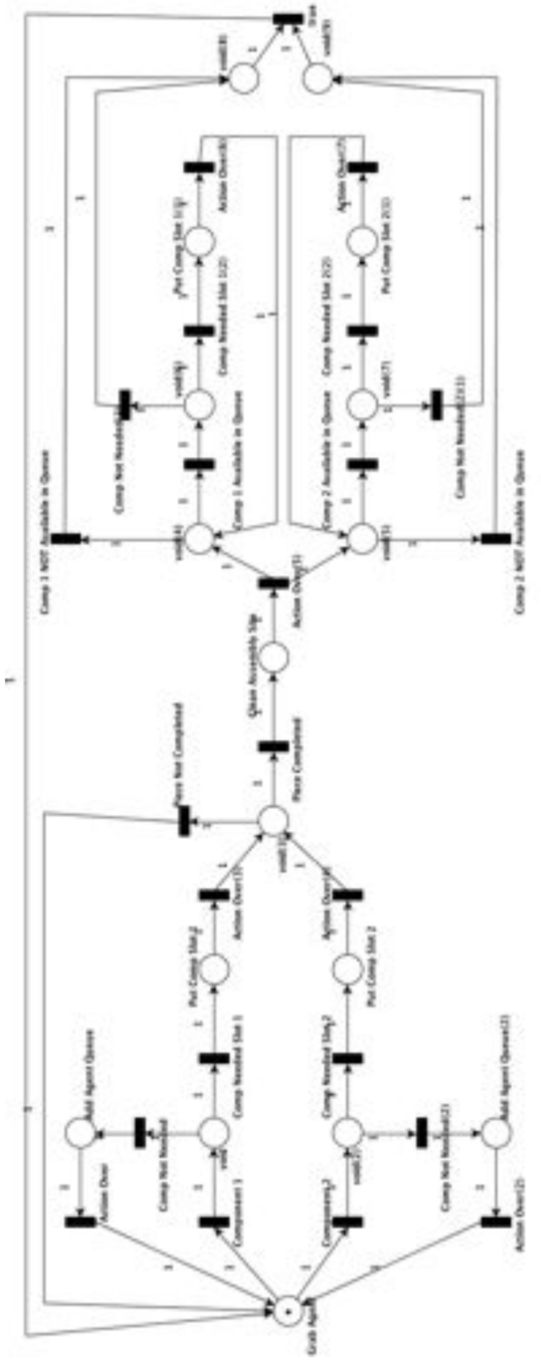


Figure 8.4: EPN controller executed by the institutional assembler in the institutional approach. Controller designed for environments where $C = 2$ and $S = 2$.

the “deliver to institutional assembler” action. This means that instead of placing their collected components directly at the assembly site, they deliver them to the institutional assembler that choses the correct slot for the component. An additional institution for role recognition could be considered but, since role recognition has already been studied in the previous case study, we chose to simplify the controller.

The robot assigned to the institutional assembler role executes a different EPN, displayed in Fig. 8.4. The controller showed is designed specifically for an environment where $C = 2$ and $S = 2$. Despite that, the design of the controller for different environments (with different C and S) is modular and easily obtained. A controller for an environment where $C = 4$ and $S = 10$ is displayed in Fig 8.5. The inherent modularity is easily observed, with the number of modules necessary being a function of C and $\lfloor S/C \rfloor$. The total number of places of the EPN is $3 + C * (5 + 2\lfloor S/C \rfloor)$ and the total number of transitions is $4 + C * (6 + 4\lfloor S/C \rfloor)$.

The institutional assembler grabs the component delivered to it by a nearby agent and checks if the component is needed given the current configuration of the assembly site. If not, the robot is put in the queue. Otherwise, the component is placed in the correct slot. After each placement the institutional assembler checks if a new piece has been completed. If so, it clears the corresponding slots in the assembly site and checks the queue for necessary components for a new piece, releasing the appropriate agents from the queue. The institutional assembler executes this controller for every component delivered to it.

8.7 Conflicts of Interest and Heterogenous Population

The social dilemma present in this task is composed by two different levels of conflicting interests. First, robots must compete among themselves in order to place components in slots of higher index and thus obtain a higher immediate reward, allowing them to remain operational for some more time. This represents a conflict of interests between individuals, since not every robot will have access to their desired (higher index) slots.

Second, robots must decide whether to place the component in a slot that gives them a higher immediate reward or a slot that allows a piece to be completed (immediately or in the future). The latter allows robots to obtain a collective reward, even if that reward comes only later in time and, in fact, may not come at all due to the behavior of other robots. This represents a conflict of interests between the individual robot and the robot team. On the one hand, the action that gives a higher immediate reward to the robot might not be the one that is best in order to achieve the overall collective goal

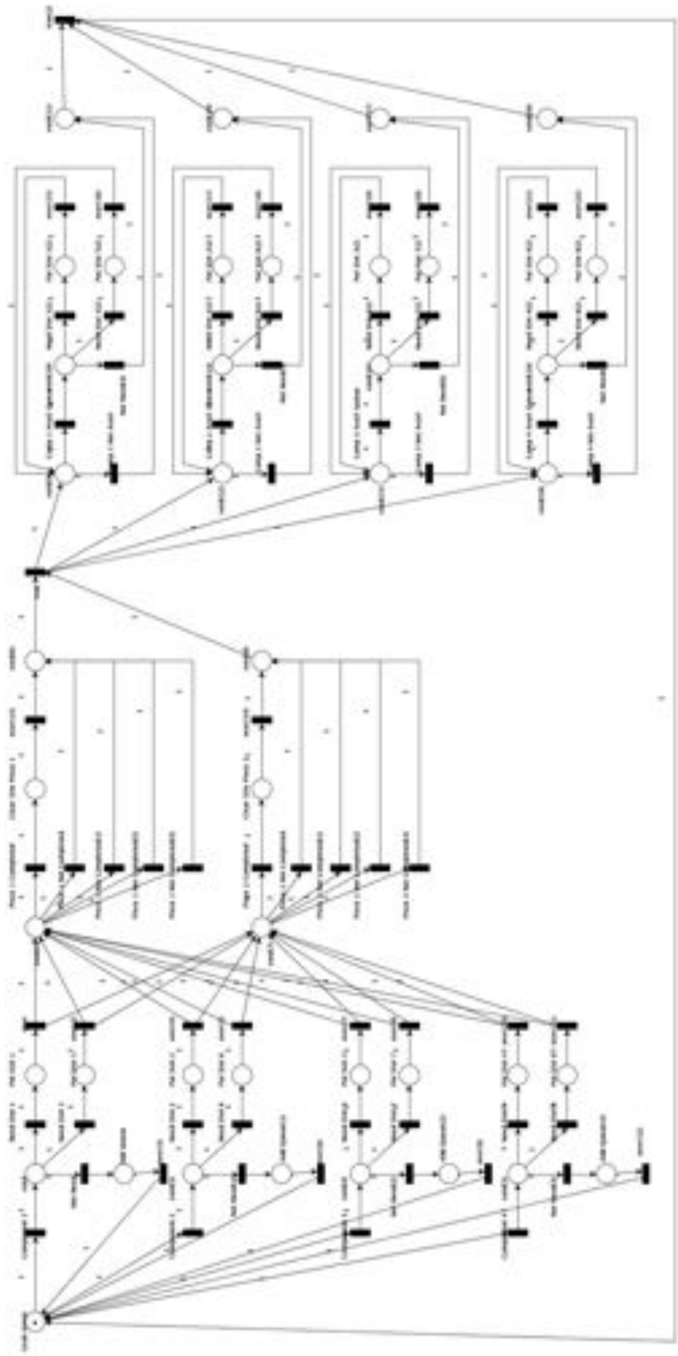


Figure 8.5: EPN controller executed by the institutional assembler in the institutional approach. Controller designed for environments where $C = 4$ and $S = 10$.

of assembling pieces; on the other hand, without a high immediate reward the robot has more difficulty in achieving its individual goal of remaining operational.

These conflicts of interest are impacted (possibly resolved) by the temporal perspective the robots have of their future rewards. When delivering a collected component to the assembly site, robots assess what rewards can be obtained at that point and in the future. If more importance is given to rewards obtained sooner rather than later, they will prefer obtaining higher immediate rewards, reflecting a tendency to assure their continued operation before considering the collective goal. But if later rewards are considered to be more important, they might accept lower immediate rewards in order to obtain a larger reward in the future, pursuing the collective goal but possibly increasing the chances of becoming non-operational.

In Section 8.4 we described how robots take into account a discount factor $\gamma \in [0, 1]$ when calculating possible future rewards coming from pieces being completed. By weighing only the contribution of collective rewards (as opposed to also affecting immediate rewards), the discount factor γ effectively reduces their importance when robots make the component placement decision. The lower γ is, the more reduced will be the contribution from collective rewards, with the extreme case of, when $\gamma = 0$, only immediate rewards being considered when placing components at the assembly site. We can say that the γ parameter represents the robots' internal motivations about collective goals versus individual goals.

In economics, discount rates are used to consider the future benefits agents can obtain from certain actions. Low discount rates yield low benefits and high discount rates yield high benefits. In [Ostrom, 1990], the author studies how agents with different discount rates affect the sustainability of resource-dependent populations. An example given is of a fishery, where local fishers living in nearby villages have low discount rates, choosing low benefits in the short term, hoping to preserve resources for future generations. On the other hand, more mobile fishers have higher discount rates, fully exploring the available resources for higher benefits, since they are not concerned with sustainability. We can say that agents with low discount rates consider long time horizons and agents with high discount rates consider short time horizons, inversely to what happens with the γ parameter in our robots.

Following this inspiration, we consider two types of robots in our population, which we will designate by *short-sighted* and *far-sighted*. Short-sighted robots are more concerned with remaining operational and prefer to obtain higher immediate rewards. Far-sighted robots give more importance to completing pieces and obtaining collective rewards. The only difference in the specification of these robots is that short-sighted robots have a low discount factor ($\gamma_S = 0.1$) while far-sighted robots have a high discount factor ($\gamma_F = 0.9$). The proportions of these types of robots in the population will be one of the parameters we will study in our microscopic model of the case study. Our intuition is that an increasing proportion of short-sighted robots will lead to an unsustainable system.

Heterogeneity in the system comes not only from the existence of different types of robots, but also from the existence of different goals. Far-sighted robots all pursue the same collective goal of completing pieces. Thus, in a population consisting only of far-sighted robots, there is only one goal being pursued by the robot team. However, this is not the case when short-sighted robots are present in the population. Short-sighted robots all purse their individual goals of remaining operational, but these goals are not the same. The individual goal of short-sighted robot i to remain operational is different from the individual goal of short-sighted robot j to remain operational. Moreover, these individual goals are conflicting, as an action that allows robot i to achieve its individual goal might prevent robot j to achieving its own individual goal (since not every robot will be able to get high rewards needed to remain operational). An increase in the proportion of short-sighted robots in the population leads to an increase in the number of individual goals being pursued in the system. Even when the population is fully comprised of short-sighted robots, resulting in a homogenous robot team, there is a heterogeneous set of goals being pursued. Far-sighted robots still have individual goals, however, they always give more priority to the shared collective goal, even if that policy causes them to become non-operational.

It must be noted that this heterogeneity in the population is only relevant when testing the decentralized approach, when robots must make the component placement decision by themselves. In the institutional approach, the responsibility of placing components at the assembly site is transferred to the institutional assembler. Since robots do not have to choose between individual or collective goals, all robots behave exactly in the same manner.

8.8 Experimental Setup

The simulated environment is toroidal and has an area A of 50×50 *units*. We consider the number of slots and number of component wells fixed to $S = 10$ and $C = 4$. When discussing the probabilistic modeling of this case study in a latter section we will also discuss a simpler instantiation of the case study slots and components.

For microscopic simulations with the decentralized approach we vary the proportion of short-sighted robots (SSR) in the population from 0% to 100%. This will allow us to observe the impact of having a heterogenous population and an increasing number of robots more inclined to fulfill their individual goal, rather then the collective goal. As mentioned previously, this parameter has no impact on the institutional approach. In this approach we vary the percentage of the collective reward that the institutional assembler takes as the assembler fee (AF) from 0% to 90%. This parameter allows us to study what costs for institutions are acceptable to the system before it becomes unsustainable.

We also vary three other parameters in microscopic simulations, both for decentralized and institutional approaches: the number of robots, the collective reward, and the simulation time horizon. We consider different values for collective reward ($CR \in \{50, 100, 150, 200\}$), in order to represent different weights given to the collective goal over the individual goal. For the simulation time horizon, the total number of time steps for which the simulation is performed, we also consider different values ($T \in \{5000, 10000, 15000, 20000, 25000, 100000\}$). Varying this parameter allows us to observe how our system evolves over transient and steady-states phases. Considering different numbers of robots ($N \in \{10, 50, 100, 1000\}$) allows us to observe the impact of the team size.

For each different set of parameters we perform 100 runs of the microscopic simulations (only 20 runs when $N = 1000$). During those runs we collect data from each individual robot - how many time steps it was operational, how much energy it spent, what reward it obtained - and from the environment - number of pieces completed and the time in which they were completed. Table 8.2 displays the values used for experimental parameters.

8.9 Results

In this section we report the results of the microscopic model, implemented with both decentralized and institutional approaches to the piece assembly task. We are interested in several metrics for this case study: sustainability, efficiency, and rate of piece completion (already discussed Section 8.3). All results shown following are normalized with respect to the upper bound of the concerning metric. A detailed discussion about how the results relate to our goals will be given in Section 8.12.

8.9.1 Decentralized Approach Results

Our first objective is to assess how heterogeneity in a multi-robot system can affect its sustainability and efficiency when performing a given task. To do so we performed microscopic simulations with robot teams following the decentralized approach. In this approach, heterogeneity in the population of robots affects the performance of the team, since different types of robots make different decisions regarding component placement. As said previously, we vary the proportion of short-sighted robots (SSR) in the population

Variable	Description	Value
A	Area of the environment	50 x 50 *units*
S	Number of slots at assembly site	10
C	Number of components	4
D	Distance between assembly site and component wells	20 *units*
r_d	Detection radius for assembly site and component wells	2 *units*
e	Energy consumed per time step by one robot	0.16
V_m	Maximum velocity of the robots	0.5 *units/time-step*
γ_S	Discount factor for short-sighted agents	0.1
γ_F	Discount factor for far-sighted agents	0.9
SSR	Proportion of short-sighted robots in population	0 : 0.1 : 1
AF	Assembler fee	0 : 0.1 : 0.9
CR	Collective reward	50 : 50 : 200
IR	Immediate reward	$1 : S$
T	Simulation time horizon (in time steps)	$5k : 5k : 25k, 100k$
N	Number of robots	$10, 50, 100, 1000$

Table 8.2: Experimental parameters value table.

from 0% to 100%, leading to a heterogenous population, not only in terms of different types of robots, but also in terms of different goals being pursued.

In Fig. 8.6-(a), 8.7-(a), and 8.8-(a), we display the sustainability, efficiency, and rate of piece completion, average values obtained in microscopic simulations performed with robot teams following the decentralized approach, for fixed $N = 50$, $T = 25000$, and $CR = 200$. We are able to observe how each metric evolves as the proportion of SSR in the population is increased. Sustainability is at maximum value for populations with less than 60% proportion of SSR and drops as this parameter is increased. Both efficiency and rate of piece completion have relatively low values (when compared to upper bound) at 0% proportion of SSR that further decrease as more SSR are present in the population. For comparison, we also plot the average values obtained in microscopic simulations with robot teams following the institutional approach and different values of assembler fee $(0, 50, 90\%$ of collective reward taken as assembler fee). These values are presented as horizontal lines since they are independent of the proportion of SSR in the population.

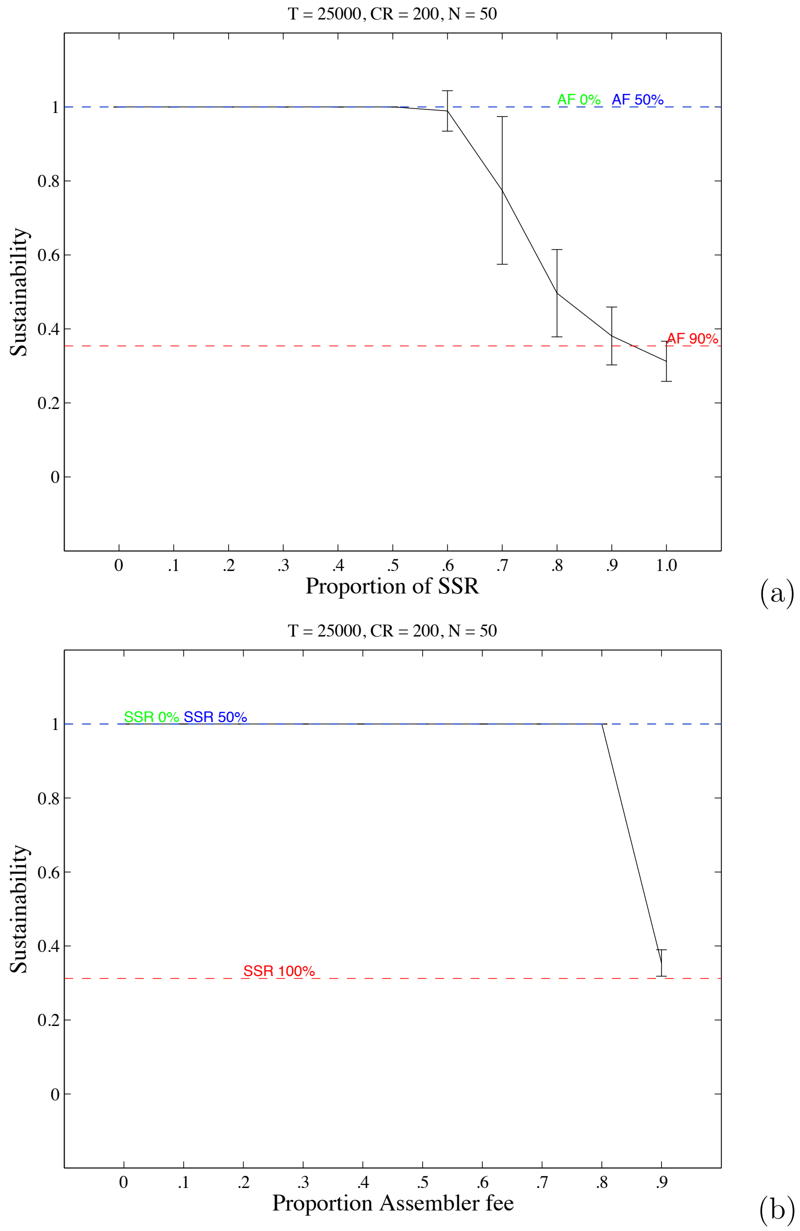
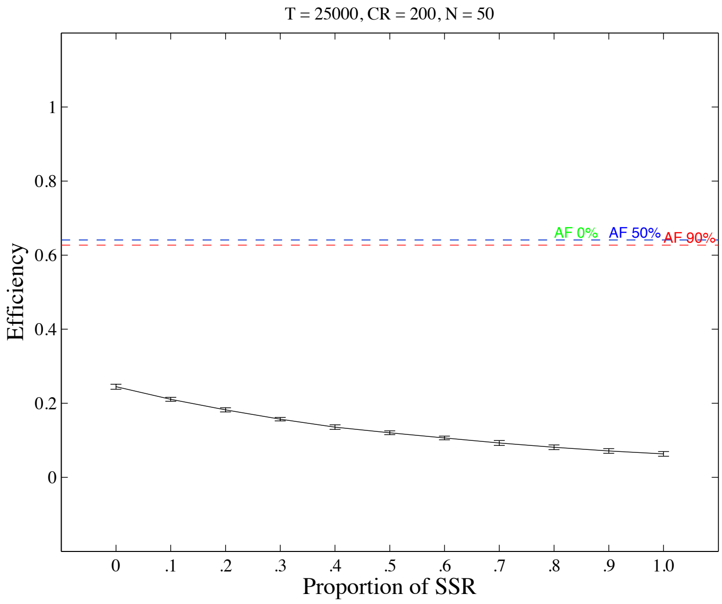
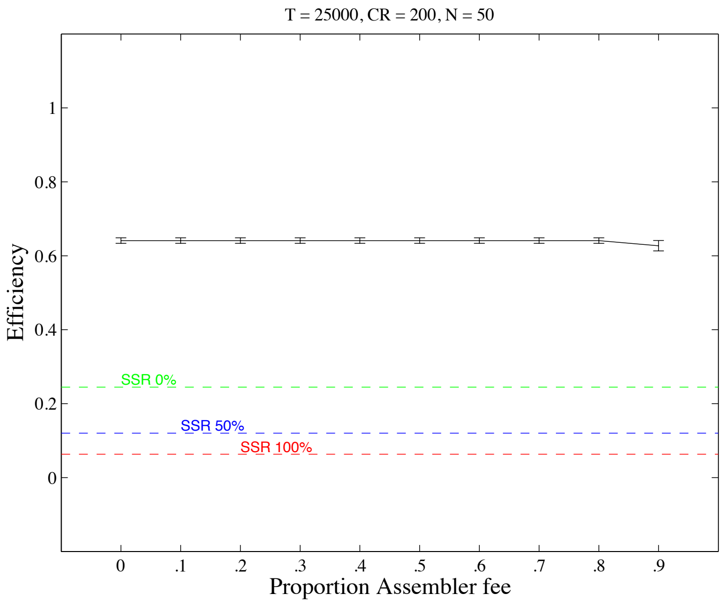


Figure 8.6: Average sustainability, bars represent variance: (a) decentralized approach with different proportions of SSR, horizontal dashed lines represent values for institutional approach, green 0% AF, blue 50%, red 90%; (b) institutional approach with different values of AF, horizontal dashed lines represent values for decentralized approach, green 0% SSR, blue 50%, red 100%.



(a)



(b)

Figure 8.7: Average efficiency, bars represent variance: (a) decentralized approach with different proportions of SSR, horizontal dashed lines represent values for institutional approach, green 0% AF, blue 50%, red 90%; (b) institutional approach with different values of AF, horizontal dashed lines represent values for decentralized approach, green 0% SSR, blue 50%, red 100%.

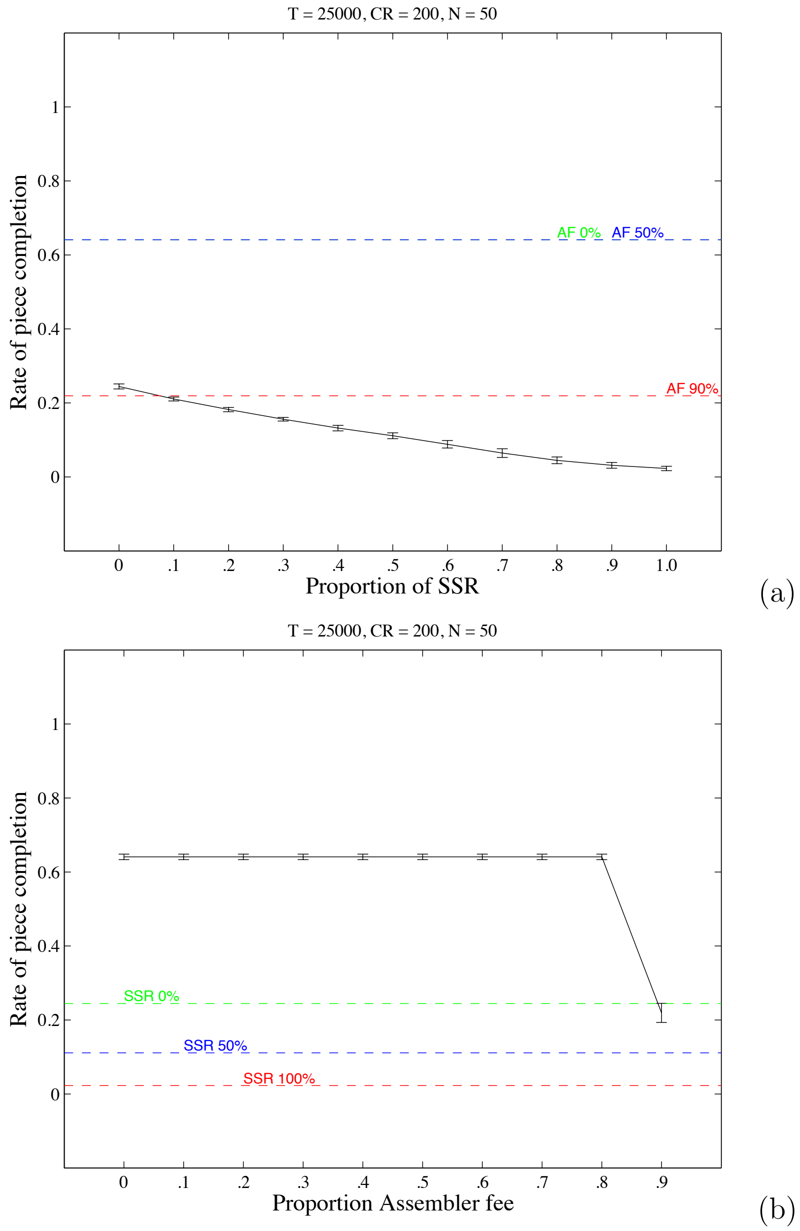


Figure 8.8: Average rate of piece completion, bars represent variance: (a) decentralized approach with different proportions of SSR, horizontal dashed lines represent values for institutional approach, green 0% AF, blue 50%, red 90%; (b) institutional approach with different values of AF, horizontal dashed lines represent values for decentralized approach, green 0% SSR, blue 50%, red 100%.

8.9.2 Institutional Approach Results

Our second objective is to assess what impact can the introduction of an institution have on the sustainability and efficiency of system. To some extent, this can already be observed in Fig. 8.6-(a), 8.7-(a), and 8.8-(a). For some values of assembler fee, the institution makes the system sustainable and more efficient for all proportions of SSR in the population. Nevertheless, costs associated with this approach must be taken into account. Our third objective is to assess what institutional costs are acceptable without becoming too heavy for the system to be sustainable and efficient. In order to answer these questions, we performed microscopic simulations with robot teams following the institutional approach. We vary the proportion of the collective reward taken as assembler fee (from 0% to 90%) to represent different institutional costs.

In Fig. 8.6-(b), 8.7-(b), and 8.8-(b), we display the sustainability, efficiency, and rate of piece completion, average values obtained in microscopic simulations performed with robot teams following the institutional approach, for fixed $N = 50$, $T = 25000$, and $CR = 200$. We observe how each metric evolves as the proportion of collective reward taken as assembler fee is increased. We also plot the average values obtained in microscopic simulations with robot teams following the decentralized approach and different proportions of SSR in the population $(0, 50, 100\%)$. These values are presented as horizontal lines since they are independent of the assembler fee considered. We observe that efficiency is also independent of the assembler fee. However, sustainability and rate of piece completion drop considerably once more than 80% of collective reward is taken as assembler fee.

8.9.3 Impact of Experimental Parameters

In all the results displayed above we have maintained fixed the number of robots, simulation time horizon, and collective reward. In order to assess what impact these parameters have in our metrics of interest we separately vary one of them keep the others fixed. In Fig. 8.9-(a) and 8.9-(b), we display the sustainability and efficiency average values obtained in microscopic simulations performed with robot teams following the decentralized approach and the institutional approach for different simulation time horizons and fixed $N = 50$ and $CR = 200$. We observe that the simulation time horizon has no impact on efficiency of the system. Moreover, we observe that, as expected, this parameter has an impact on the sustainability values. However, for both approaches, it is noticeable the existence of threshold values, after which the sustainability decreases.

In Fig. 8.10-(a) and 8.10-(b), we display the sustainability and efficiency average values obtained in microscopic simulations performed with robot teams following the decentralized approach and the institutional approach for different collective reward values and fixed $N = 50$ and $T = 25000$. It is noticeable that there are values of collective reward too low for the system to perform as expected ($CR = 50$). We observe that efficiency remains constant. In the sustainability plot we observe that with the increase of the collective reward there is an increase of the sustainability threshold discussed above.

In Fig. 8.11-(a) and 8.11-(b), we display the sustainability and efficiency average values obtained in microscopic simulations performed with robot teams following the decentralized approach and the institutional approach for different numbers of robots and fixed $CR = 200$ and $T = 25000$. Sustainability is close to constant with different numbers of robots. However, we can observe that, in the institutional approach, efficiency increases with the increase in team size. This is not observed in the decentralized approach. We will discuss this and previous results in Section 8.12.

8.10 Adaptation to Changing Environments

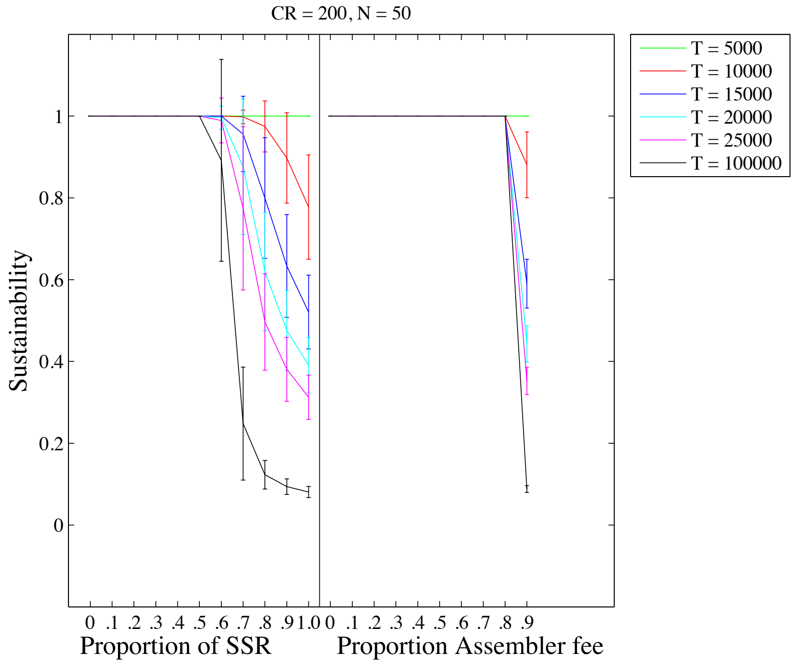
In this section we extend our piece assembly case study to consider different environments with different locations for component wells. This will allow us to study and compare the robustness of our decentralized and institutional approaches.

Considering environments where component wells are not all at the same distance of the assembly site has two main implications. First, the rate of components arriving at the assembly site is different for different component types. Second, the energy spent by the robots collecting and delivering components to the assembly site varies with the type of component.

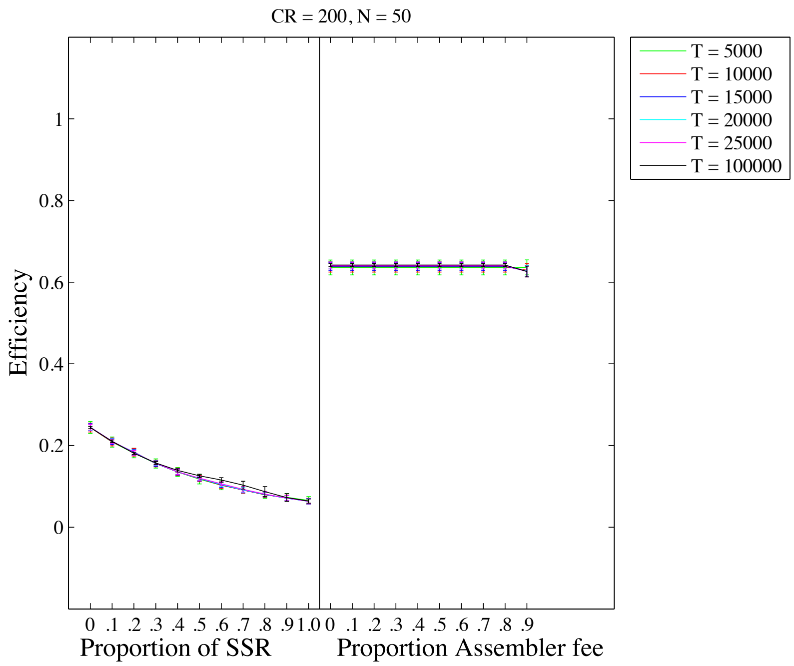
The former implication can have a major impact in the performance of our system. So far, the system has operated under the assumption that all component types are equally available. However, if one (or several) of the component types is predominant over other types, the effort needed to assemble pieces will be increased. The latter implication will have an impact in the sustainability of the system, as robots will need more or less energy to travel to wells that are further or closer than the previously considered distance.

8.10.1 Perturbations to the Environment

In the previously described environment, we considered C component wells located in a circle of radius D around the assembly site. In this set of microscopic simulations, we will consider perturbations that move the component wells from those locations. However, we will restrict ourselves to perturbations that move one or two wells further or closer away, while maintaing their relative orientation towards the assembly site.



(a)



(b)

Figure 8.9: Average sustainability (a) and average efficiency (b) for decentralized approach (with different proportions of SSR, left side) and institutional approach (with different assembler fees, right side) for different simulation time horizons and fixed $N = 50$ and $CR = 200$

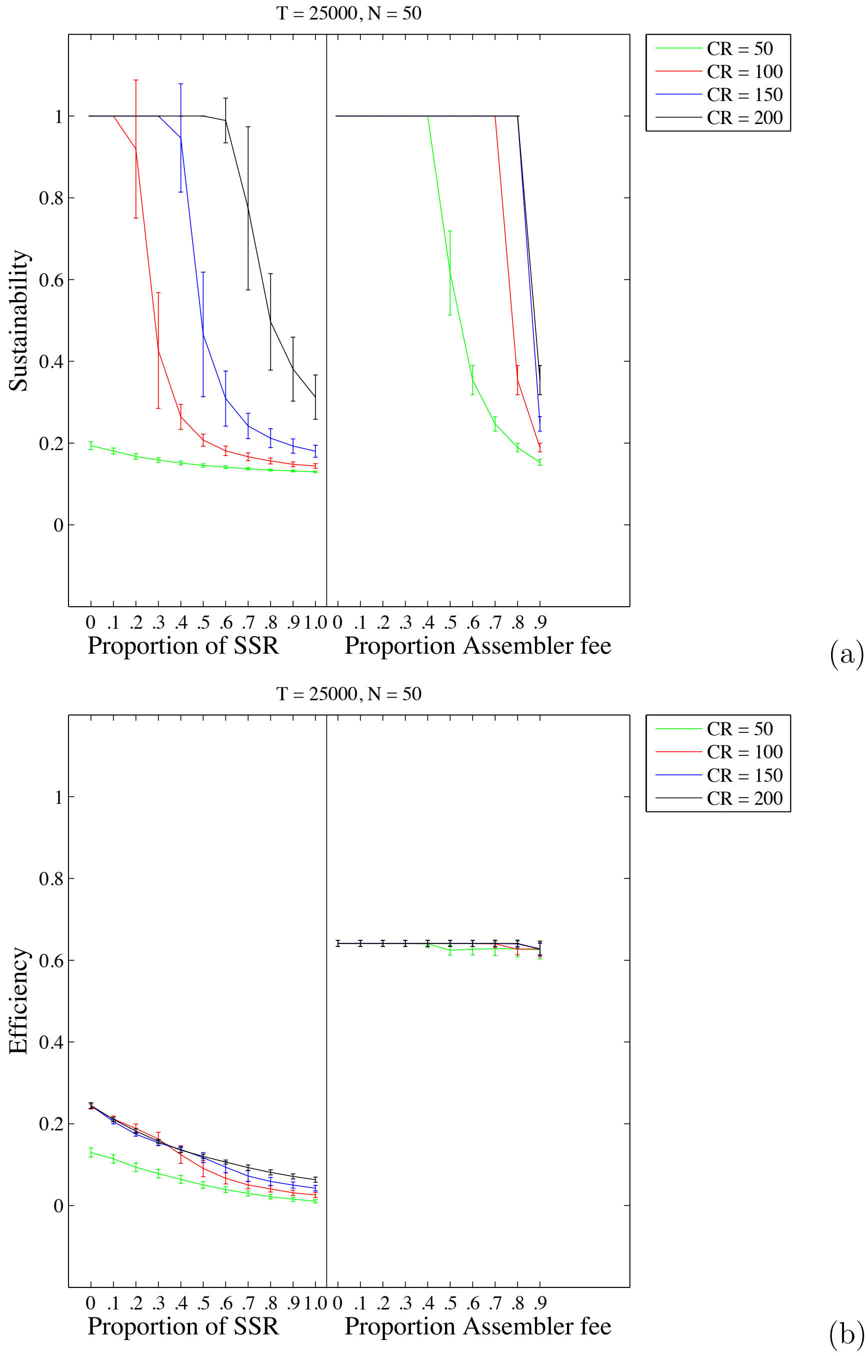
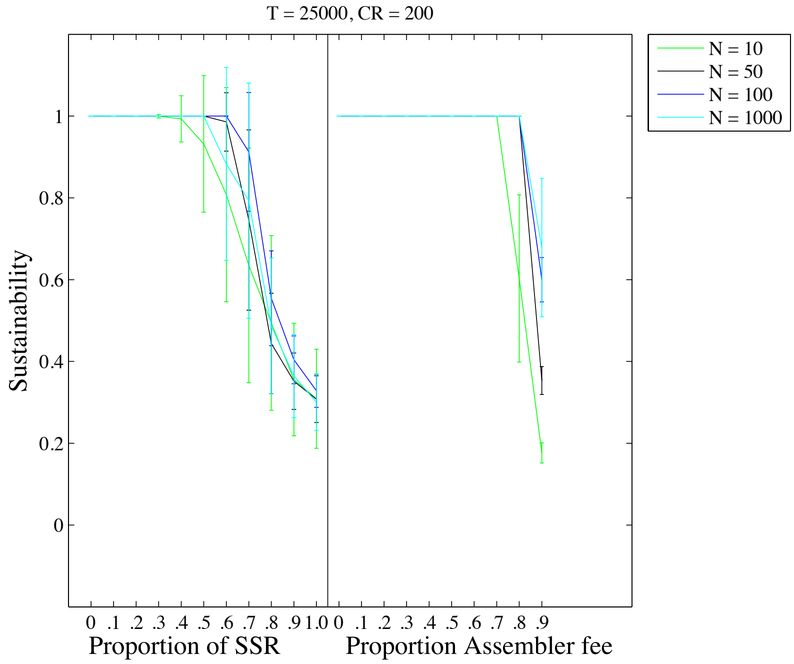
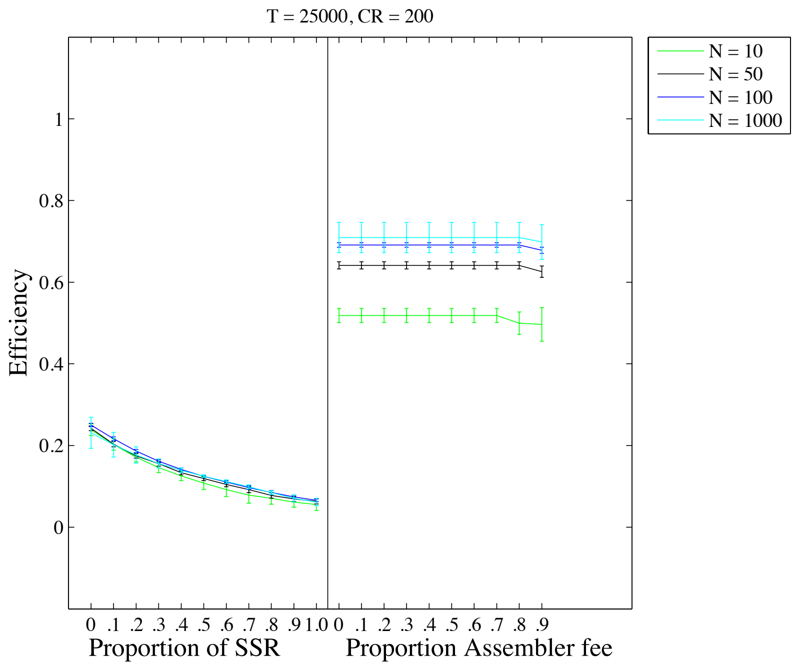


Figure 8.10: Average sustainability (a) and average efficiency (b) for decentralized approach (with different proportions of SSR, left side) and institutional approach (with different assembler fees, right side) for different collective reward values and fixed $N = 50$ and $T = 25000$.



(a)



(b)

Figure 8.11: Average sustainability (a) and average efficiency (b) for decentralized approach (with different proportions of SSR, left side) and institutional approach (with different assembler fees, right side) for different number of robots and fixed $T = 25000$ and $CR = 200$.

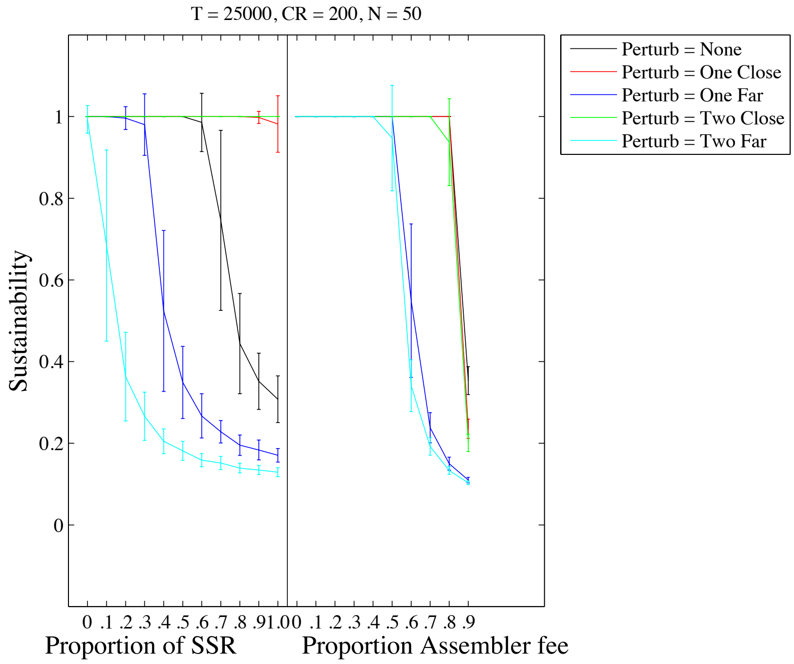
Despite restricting a wider range of possible locations for component wells, for instance randomly placing them throughout the environment, the perturbations considered accurately reflect the two main implications to the case study mentioned previously. The rate of components arriving at the assembly site varies with the distance between component wells and assembly site. This arises from how the probabilities of choosing the next well to visit are calculated by each robot. Following our intuition that these probabilities decrease as distance increase, we will have a proportional decrease in rate of components arriving with the increase of distance. The energy spent collecting and delivering components to the assembly site will also vary with distance of component wells. Previously, we saw that the value e of energy consumed per time step by the robots was obtained using the distance D between assembly site and component wells. We leave e unaltered irregardless of the perturbations made to the environment.

We perform microscopic simulations with four different types of perturbation: i) moving one component well closer to the assembly site; ii) moving one well further away; iii) moving two wells closer; iv) moving two wells further away. Component wells moved closer are located at distance $D' = D/2$ of the assembly site, while wells moved further away are located at distance $D'' = 2D$. During these microscopic simulations we leave the number of robots, collective reward, and simulation time horizon, fixed to $N = 50$, $CR = 200$, and $T = 25000$, and increase the area of the environment to 100×100 units to accommodate for D''. We perform 100 runs of the microscopic simulations for each of the perturbations, both with robot teams following the decentralized and the institutional approaches. As before, for microscopic simulations with the decentralized approach we vary the proportion of short-sighted robots in the population, and for microscopic simulations with the institutional approach we vary the proportion of the collective reward taken by the institutional assembler as assembler fee. The upper bound for the rate of piece completion is obtained considering $D = 20$ as in the previous set of microscopic simulations.

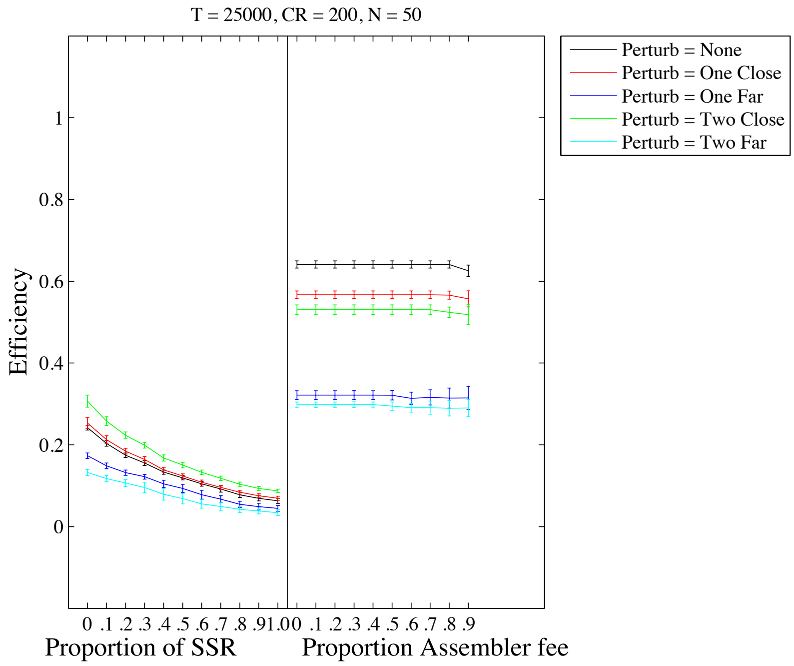
8.10.2 Results

In Fig. 8.12-(a), 8.12-(b), and 8.13, we display the sustainability, efficiency, and rate of piece completion, average values obtained in microscopic simulations performed with robot teams following the decentralized and the institutional approaches for different perturbations to the environment.

In microscopic simulations with the decentralized approach we observe that perturbations have the same effect in all three metrics. Perturbations that move components wells further away from the assembly site have a prejudicial effect, while perturbations that move component wells closer to the assembly site actually improve the performance. As



(a)



(b)

Figure 8.12: Average sustainability (a) and average efficiency (b) for decentralized approach (with different proportions of SSR, left side) and institutional approach (with different assembler fees, right side) for different perturbations to the environment.

before, the decline of each metric with the increase in the proportion of short-sighted robots in the population is maintained.

However, this effect is not reproduced in microscopic simulations with the institutional approach. We observe that all perturbations to the environment have a prejudicial effect on our metrics of interest, irregardless of moving component wells closer or further away. While this effect is small for sustainability values of microscopic simulations where components wells are closer to the assembly site (Fig. 8.12-(a)), it is clearly observable in the values of efficiency and rate of piece completion (Fig. 8.12-(b) and 8.13). Despite this, we observe that the evolution of each metric with the increase in assembler fee is maintained. Efficiency remains constant, while sustainability and rate of piece completion have sharp declines once a critical assembler fee is reached.

The prejudicial effect the perturbations have in microscopic simulations with the institutional approach is caused by how the institutional assembler deals with delivered components that are not needed at delivery time. Robots holding such components are held in a queue, from which they leave once their collected component is needed or when a timeout is reached. Moreover, this timeout is not dependent on the type of component the robot is holding. All the considered perturbations to the environment create a certain predominance of one (or more) type of component over the others, causing robots delivering that type of components to be held in the queue more often. This increase of robots in the queue creates a decrease in the number of robots collecting and delivering the components, which explains the prejudicial effect the perturbations have in all metrics.

The different response to perturbations by the decentralized and institutional approaches is not unexpected. By distributing the task by the entire robot team, the decentralized approach becomes more robust to any changes made not only in the team, but also in the environment. Because of this, not only a decrease in rate of components arriving at the assembly site is accommodated, but also an increase actually improves performance. On the other hand, the institutional approach centralizes the piece assembly in a single agent, which improves performance over the decentralized approach but at the cost of becoming less robust to any changes. Any change that breaks the balance of rate of arriving components for different components types will be prejudicial to the system, unless the controller of the institutional assembler is changed to reflect that change. Nonetheless, it should be noted that this effect cannot be fully extrapolated to all institutional approaches since it arises from how the institutional assembler deals with components not needed at a particular time.

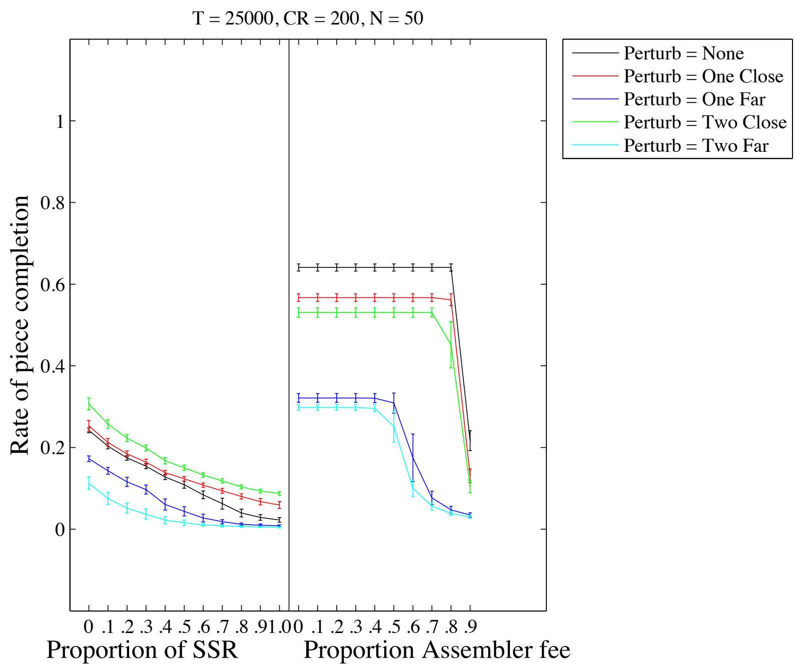


Figure 8.13: Average rate of piece completion for decentralized approach (with different proportions of SSR, left side) and institutional approach (with different assembler fees, right side) for different perturbations to the environment.

8.11 Piece Assembly Macroscopic Modeling

In this section we present a probabilistic macroscopic model for the piece assembly case study. Since the case study has so far been implemented only using microscopic simulations, our model will capture the "reality" of such simulations, allowing us to obtain quantitative predictions of the performance of the system in different situations, without the computational effort of running multiple complex simulations.

We focus on obtaining a macroscopic model of the performance (measured as the rate of piece completion) of the institutional approach to the piece assembly case study. The performance is modeled under the set of perturbations to the environment described in the previous section. This allows us not only to observe if it is possible to capture more complex tasks in an institutional setting with macroscopic models, but also if those models are able to capture different responses of the system to different environments.

We perform a new set of microscopic simulations with the institutional approach for teams of $N = 10$ robots under different perturbations to the environment. We consider this lower number of robots to keep the model complexity low, since this N is used as an environmental information parameter in the macroscopic model and its increase generates an increase in state space (as will be seen further ahead). Note that N is used solely as a parameter, we do not mode single robots individually. As before, microscopic simulations are conducted in an environment with $S = 10$ and $C = 4$. No energy is considered. For this section we also conducted microscopic simulations in a simpler environment with $S = C = 2$. For all simulations we store the rate of components arriving at the assembly site. The stored rates will be used as a "ground truth" comparison for analytically estimated rates that will be used as input parameters to the macroscopic model.

8.11.1 Model Structure

As discussed in Chapter 5, our GSPN macroscopic model uses, as a starting point, the EPN structure of the IAC designed for the task. Information about piece completion is only present on the EPN controller executed by the institutional assembler, represented by the transition "Piece Completed". Since we are interested in modeling only the rate of piece completion we will use this EPN, displayed in Fig. 8.4, as starting point. This correspond to using the IAC lower layer implementation of the institution. Our goal is to be able to estimate the throughput of transition "Piece completed", representing the rate of piece completion.

In order to transform the EPN controller into a GSPN model, the first step is to identify which transitions correspond to timed transitions. The remaining transitions will be considered as immediate. In our simulations, most actions performed by the institutional assembler have an immediate effect, meaning that they occur in a single time step. The only transitions in the controller that do not have this property are those concerning the arrival of components to the assembly site. Thus, the only timed transitions in our GSPN will be those concerning the arrival of components, in the case of Fig. 8.4, transitions "Component 1" and "Component 2". Transition rates for such transitions must be estimated and will be the input parameters of the model.

Piece completion is also verified by the institutional assembler in a single time step, but we will consider the transition as timed (although with a very small rate, several orders of magnitude lower than other transition rates). We do this to facilitate the analysis of the GSPN model. Measuring throughput of immediate transitions is possible but follows a complex algebraic process that, by requiring a matrix inversion, increases in complexity with the increase of the state space of the GSPN [Bause and Kritzinger, 2002].

There is not sufficient environmental information in the GSPN to correctly model the piece assembly process. Information about the assembly site state, size of the queue (and the types of components present therein) is not represented in the controller. Thus, we need to add this information in the environmental information layer of our GSPN macroscopic model, following the methodology described in Section 5.4.

We add two boolean predicate places for each slot of the assembly, representing if the slot is free or occupied. We also add memory places representing the number of available robots (robots not in the queue) and the number of robots in the queue for each type of component. For all the added memory places we add a complement place. These will allow transitions to check if a place is *not* marked without the need for inhibitor arcs. We consider robots timing out when in the queue as an environmental process and add timed transitions to represent it (connected to the memory places). Connections between transitions in the controller and places in the environmental information layer are added depending on whether transitions are action finalizers or information checkers. In the lower section of Fig. 8.14 we display the places and transitions added to the environmental information layer, and exemplify connections to two transitions. The "Put Comp Slot 1" transition actually changes the state of the environment, while the "Piece Completed" transition just checks what this state is.

The full GSPN macroscopic model is obtained by connecting all transitions to the relevant environmental information, as can be seen in Fig. 8.15 for the case of an environment with $S = C = 2$. We also construct such a model for the larger environment ($S = 10$ and $C = 4$).

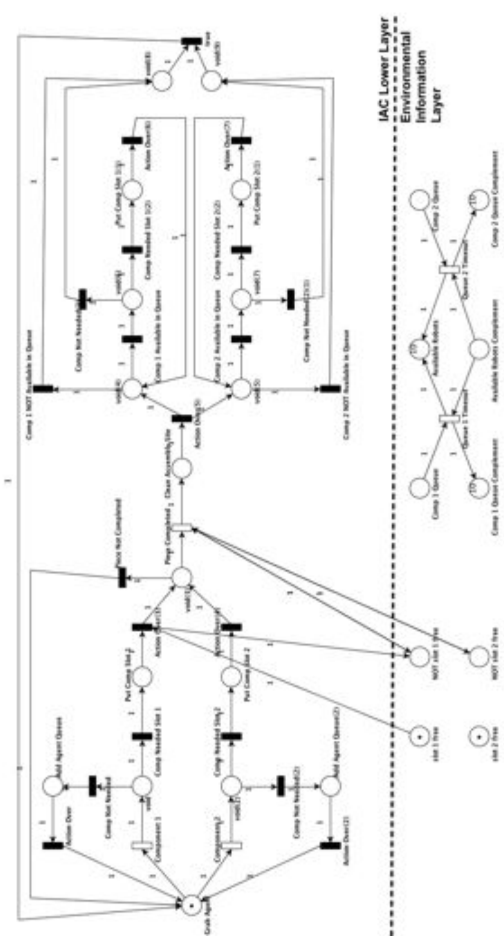


Figure 8.14: Intermediate stage in GSPN macroscopic model construction. Model designed for environments where $S = C = 2$. Timed transitions are represented as empty rectangles and immediate transitions as full rectangles.

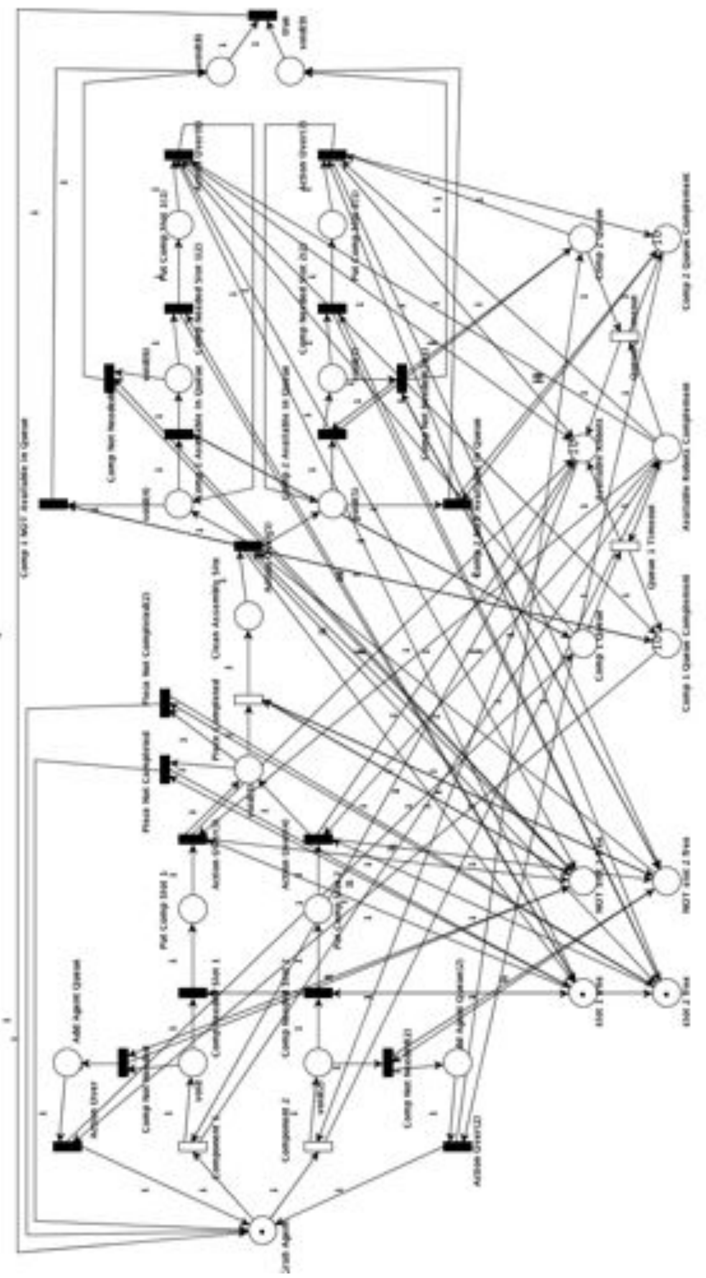


Figure 8.15: Full GSPN macroscopic model for the piece assembly case study. Model designed for environments where $S = C = 2$. Timed transitions are represented as empty rectangles and immediate transitions as full rectangles.

8.11.2 Rates Estimation

As mentioned previously, the rates of component arrival at the assembly site will be used as input parameters for our GSPN macroscopic model. They are dependent on which environment we consider. Our goal is to obtain an a priori model of the system, capable of being tested before performing any simulation or real robot test. To do so, we must obtain an estimate of the rates of component arrival. This estimation will be obtained analytically based on the geometrical properties of the environment.

However, these rates can also be obtained from microscopic simulations data. These data-obtained rates are useful to provide some "ground truth" when verifying our estimates and, if used as parameters for our GSPN model, should provide more precise estimations for rate of piece completion. Nevertheless, we will not rely on them for our GSPN macroscopic model, as that would force us to perform microscopic simulations to obtain transitions rates before being able to analyze the GSPN.

Let r_i be the real rate of components of type i arrival at the assembly site. We designate as $\tilde{r}_i$ the estimation of r_i obtained from the simulation data. We store the number of components delivered to the assembly site during one simulation as c_i and compute $\tilde{r}_i$ as $\tilde{r}_i = c_i/T$.

The analytically obtained estimation of rate r_i will be designated as $\hat{r}_i$. This rate can be seen as the inverse of a time delay $\hat{d}_i = 1/\hat{r}_i$ between arrivals of components of type i to the assembly site. This delay can be computed from the rounded number of time steps necessary to perform a round trip visit between assembly site and the i-th component well $T_{v,i}$, the number of components C, and the number of robots N. The time delay between arrivals of component i is calculated in the following manner:

$$\hat{d}_i = \frac{T_{v,i}}{N(\frac{1}{C})} \tag{8.8}$$

The time necessary to perform a round trip visit between assembly site and component is divided by the number of robots to represent the contribution of all robots, and by $1/C$ to represent the possibility of the robot going to any of the C component wells. We can now obtain our rate estimations as $\hat{r}_i = 1/\hat{d}_i$.

In Fig. 8.16 and 8.17 we display the comparison between the "ground truth" rates of component arrival at the assembly site, obtained through microscopic simulations, and our estimate of such rates, for environments with $C = 2$ and $C = 4$ respectively. We observe that our rate estimation is accurate for component wells located at the standard distance (or further) to the assembly site. However, for wells located close to the assembly site, we overestimate the rate at which components arrive at the assembly. This is likely

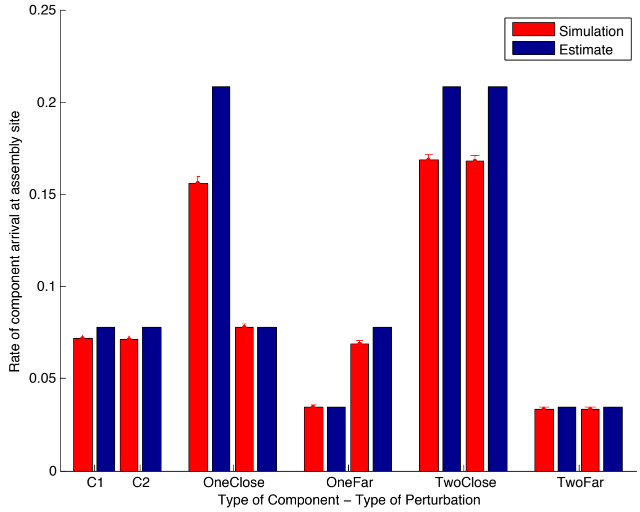


Figure 8.16: Comparison between rates of component arrival at assembly site obtained through simulation and through analytical estimation for environments where $C = 2$. Red bars represent rates $\tilde{r}_i$ while blue bars represent rates $\hat{r}_i$. The first set of bars represents an unperturbed environment (all wells at same distance), while the remaining sets of bars represent rates for different components under different perturbations to the environment.

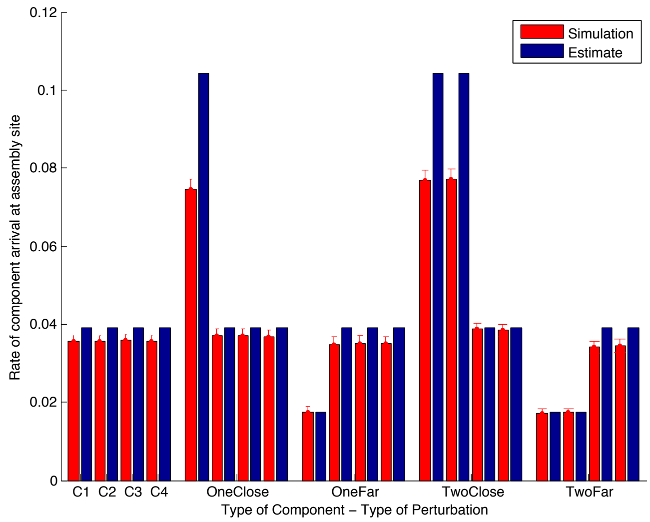


Figure 8.17: Comparison between rates of component arrival at assembly site obtained through simulation and through analytical estimation for environments where $C = 4$. Red bars represent rates $\tilde{r}_i$ while blue bars represent rates $\hat{r}_i$. The first set of bars represents an unperturbed environment (all wells at same distance), while the remaining sets of bars represent rates for different components under different perturbations to the environment.

due to the fact that more robots choose to visit the closer component wells, leading to a high density of robots in a smaller space than for other component wells. Since robots perform a basic obstacle avoidance maneuver, they take more time than what is estimated do deliver the components. We could try to obtain a better estimate of the rates of component arrival that included the obstacle avoidance behavior but we feel that the current rates are enough to obtain a good model.

8.11.3 GSPN Analysis Methods

In Section 5.2 we noted that GSPNs can be classified according to several axis dealing with the firing policy of transitions. To more accurately capture the case study being modeled herein we will consider a non-standard firing policy.

Timed transitions that can be enabled multiple times can fire according to different types of timing semantics [Ajmone Marsan et al., 1989, Ajmone Marsan et al., 1995]. A transition is enabled k times if its input places have k times the tokens needed for the transition to fire. Depending on the activity being modeled, different timing semantics may be used to deal with these multiple tokens. Two possibilities are *exclusive-server* and *infinite-server* semantics.

- Exclusive-server semantics: a firing delay is set when the transition is first enabled, and new delays are generated after the transition has fired, assuming it is still enabled. This models activities that are processed serially and is consider the standard semantics for timed transitions in GSPNs. This type of semantics was used in Section 6.6.
- Infinite-server semantics: a firing delay is set for each new set of tokens that enables the transition, even if the transition is already enabled and a firing delay is elapsing, causing the transition to be enabled concurrently with itself multiple times. For instance, assuming a transition needs two tokens in its input place to fire, a firing delay will be set when the first two tokens arrive. If while the delay is elapsing, another two tokens arrive at the input place, a new firing delay will be set that will elapse in parallel (transition enabled two times). This models activities that are processed concurrently.

Transitions that consider infinite-server semantics can be represented in the standard way by using marking-dependent rates. By multiplying the transition rate by the number k of times the transition is enabled the same properties are obtained. This also guarantees that the Markov chain correspondence is maintained.

We consider that our models for this case study use an infinite-server semantics for timed transitions. Consider transition “Queue 1 Timeout” and its input place “Comp 1 Queue” in Fig. 8.15. The firing of this transition represents one robot leaving the queue and returning to the set of available robots. Every token in the "Comp 1 Queue" place represents one robot in the queue. If we used the standard exclusive server semantics, tokens would exit that place (because of a queue timeout) with a given rate but in a sequential manner. Only after the timeout transition had fired, moving one token out of that place, would the timer for that transition start again. By using infinite server semantics, a transition can be enabled concurrently with itself multiple times, as many times as there are tokens in its input places. This better represents the parallelization happening in the queue, from where robots can enter and exit concurrently.

The goal of the analysis of our models is to obtain steady state estimates of the throughput of transition "Piece Completed". Unfortunately, the GSPN tools described in Section 2.3 do not allow us to obtain steady state metrics via EMC analysis for GSPNs using infinite-server semantics. We can however perform GSPN model simulations (using Monte Carlo methods) with these tools, that will allow us to obtain the same steady state metrics.

It must be noted that when considering analysis methods that use infinite server semantics, all input parameter rates must be divided by the number of robots N since, under that transition semantics, event rates are now related to single robot events (as opposed to team related events).

8.11.4 Results

We now present the results for the estimation of the rate of piece completion with different GSPN macroscopic models for different environments. These models are tested under a set of perturbations to the environment. In the following plots we display the actual rate of piece completion taken from microscopic simulations as ground truth together with the estimates obtained with the GSPN macroscopic models.

In Fig. 8.18 we display the results for the estimation of rate of piece completion with a GSPN model designed for an environment where $S = C = 2$. We observe that for the perturbation that moves both component wells closer to the assembly we overestimate the rate of piece completion. We argue that this effect is due to the overestimation of the input parameters. This will be investigated further ahead by considering a GSPN model using as input parameters the rates obtained from simulation data. For all other cases, our model accurately predicts the rate of piece completion obtained in simulation.

In Fig. 8.19 we display the results for the estimation of rate of piece completion with a GSPN model designed for an environment where $S = 10$ and $C = 4$. We observe that for this more complex environment, and consequently more complex GSPN model, we obtain accurate predictions of the rate of piece completion obtained in simulation. The effect obtained in Fig. 8.18 for the "TwoClose" perturbation is not replicated. We believe that

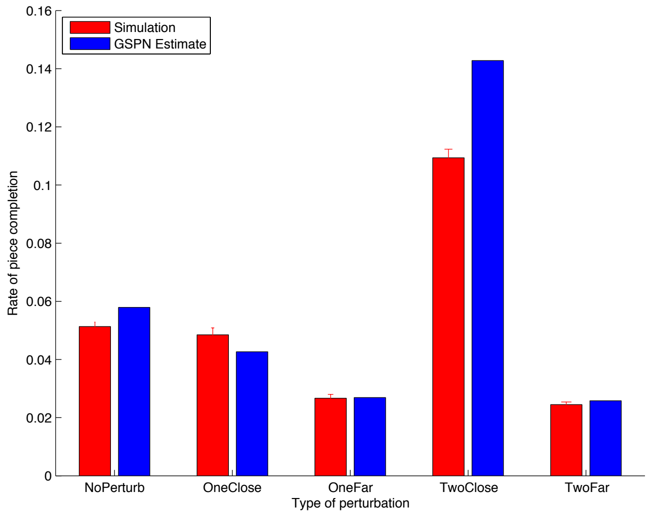


Figure 8.18: Results for estimation of rate of piece completion with GSPN model designed for an environment where $S = C = 2$, tested under a set of perturbations to the environment, and using as input parameters the analytically estimated rates of component arrival.

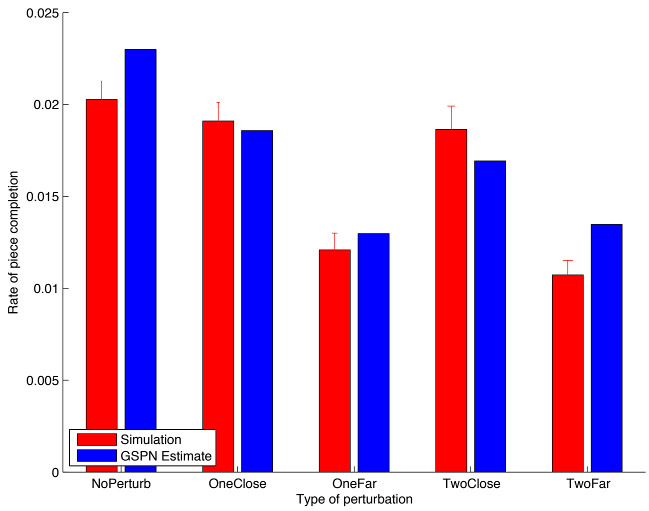


Figure 8.19: Results for estimation of rate of piece completion with GSPN model designed for an environment where $S = 10$, $C = 4$, tested under a set of perturbations to the environment, and using as input parameters the analytically estimated rates of component arrival.

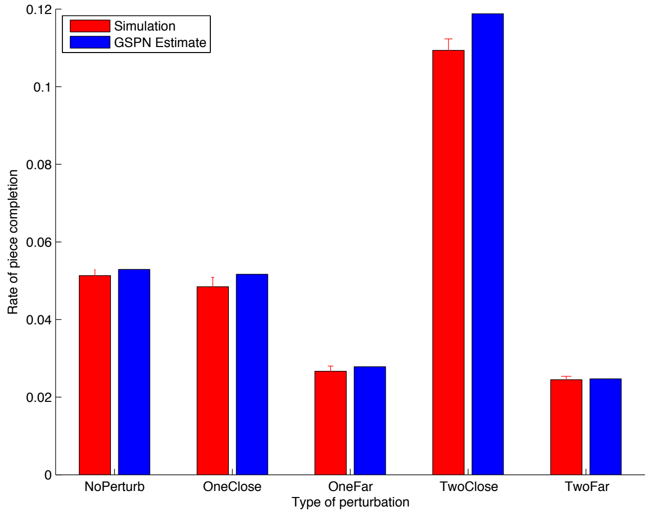


Figure 8.20: Results for estimation of rate of piece completion with GSPN model designed for an environment where $S = C = 2$, tested under a set of perturbations to the environment, and using as input parameters the simulation data-obtained rates of component arrival.

this is due to not all component wells being moved (since $C = 4$), as opposed to what happened in the case of a simpler environment.

In Fig. 8.20 we display the results for the estimation of rate of piece completion with a GSPN model designed for an environment where $S = C = 2$, but considering as input parameters the rates obtained from simulation data. Comparing with Fig. 8.18, we can observe that considering input rates from simulation data provides a better estimation than using estimated input rates.

8.12 Discussion

In this section we discuss how the results presented throughout this chapter relate to the three objectives we initially set up. These objectives deal with sustainability and efficiency of a multi-robot system, with heterogeneity within the multi-robot system, and with how an institutional approach to multi-robot systems might circumvent sustainability, efficiency, and performance, problems.

How does heterogeneity in a multi-robot system affect its efficiency and sustainability when performing a given task?

In Fig. 8.6-(a), 8.7-(a), and 8.8-(a), we observe that an increase in the proportion of short-sighted robots in the population has a prejudicial effect in sustainability, efficiency, and rate of piece completion. The system is sustainable until a critical proportion of short-sighted robots is reached (in this case 60%), from which point sustainability decreases. Efficiency and rate of piece completion decrease in a similar manner, given that efficiency is dependent on rate of piece completion. We observe that, in this case, heterogeneity does affect sustainability and efficiency.

As we have discussed before, we relate heterogeneity in the system not only with different types of robots in the population, but also with different types of goals being pursued. Short-sighted robots pursue their own individual goals while far-sighted robots pursue the collective goal. This relation between heterogeneity and different goals is not present in all instances of multi-robot systems and different case studies. Moreover, it is possible that other types of heterogeneity in the population (for instance in systems considering division of labor) can actually improve performance and efficiency. However, we believe that our case study can represent a class of distributed robotic systems where a social (or moral) dilemma is considered by the robots, and where different types of robots take different decisions regarding social dilemmas.

While this class of distributed robotic systems might not have a considerable number of examples in current research, future real world applications of these systems will undoubtedly face situations where robots will have to take decisions about pursuing individual goals or more team oriented goals. Team goals are dependent on the task the system should perform. Individual goals can be related to the robots need to maintain operation, for instance the need to recharge batteries or moving out of an area where physical damage might occur, or can be related to some urgent goal detected by a single robot. An increase in robots pursuing individual goals will possibly lead to a decrease in sustainability and efficiency.

What impact can the introduction of institutions have on the efficiency and sustainability of a multi-robot system that is not efficient and not sustainable?

Institutions can restrict behaviors that are critical for the collective performance, while decisions on the execution of other behaviors are left to each individual robot. These restrictions however should help manage the effort of team. In our case, the institutional assembler restricts other robots delivering components, in order to accomplish that task more efficiently.

One of the main ideas in the institutional robotics approach is that, if every physical agent follows the same set of rules, each robot knows how other agents will act. We use the term physical agents to describe not only multi-robot systems, but also mixed human-robot teams, where physical agents (humans and robots) can cooperate.

In distributed robotics systems where a social dilemma is present there is always a possible decision an agent can take, or a set of special circumstances the agent is in, that impairs coordination with other agents. In the institutional robotics approach, institutions are used to enforce coordination, possibly even considering penalties for agents not complying with that enforcement. By enforcing coordination, and enforcing that all agents act in the same manner by following the same set of rules, institutions can remove the social dilemmas that such distributed robotic systems face.

In Fig. 8.6-(a), 8.7-(a), and 8.8-(a), we observe the impact of the institutional assembler in the piece assembly case study (for different assembler fees). Metrics for the institutional approach are shown as horizontal lines since they are independent of the heterogeneity of the population. The institutional assembler eliminates distinctions between short- and far-sighted robots, by enforcing that all robots act in the same manner. We can also observe that not all institutions are able to improve on the decentralized approach. In this case, different institutions are represented by different costs for the institutional assembler. The institutional approach can also take advantage of larger teams of robots due to its tighter coordination. In Fig. 8.11-(b), we can observe that an increase in team size leads to an increase in efficiency, while in the decentralized approach it remains constant.

Institutions can improve sustainability, efficiency, and performance, of distributed robotic systems. However, it is possible that the tradeoff for this improvement is the loss of robustness. In Fig. 8.12-(a), 8.12-(b), and 8.13, we observe that all perturbations to the environment causes a decrease in the respective metric for the institutional approach. In our system we chose one robot to play the institutional assembler role. However, even if the robot playing the role constantly changed throughout the simulation, the burden of piece assembly would always be, at any time, on a single robot of the team. It must be noted that a better designed solution could possibly avoid such problems. However, it must be taken into account when designing institutions that, by centralizing the burden of enforcing rules for the system on a set of agents, robustness and adaptability can decrease.

What costs of these institutions are acceptable without becoming too heavy for the multi-robot system to be efficient and sustainable?

Institutions have associated costs. In our case, the institutional assembler is a robot that also needs to obtain rewards, so it has to charge an assembler fee taken from the collective reward given to robots completing a piece. These costs can be associated with more than the need of energy or reward of some robots. The design of institutions can be a time consuming process that might represent some cost for the system. The same can be said of the maintenance or monitoring of institutions. When applying an institutional approach, studies of the costs of that approach should be performed.

In our case and in Fig. 8.6-(b), 8.7-(b), and 8.8-(b), we observe that sustainability and rate of piece completion drop dramatically once a certain value of assembler fee is reached. Efficiency remains constant because, although piece completion decreases, robots become non-operational during the simulation and stop spending energy. The cost of the institutional assembler is payed by the transporting robots.

The effects of costs of institutions in our case study cannot be extrapolated to all institutions and all case studies. However, these results show that applying an institutional approach with a predesigned institution does not always improve the performance of a given distributed robotic system.

Summary

In this chapter we described a case study tackling the sustainability of distributed robotic systems. We considered a transport and assembly task, where robots have to gather components and coordinate to complete pieces. Robots spend energy while carrying out the task but obtain energy rewards when the team goal is accomplished. These energy rewards give the task some aspects of a social dilemma, where different robots give more priority to either individual or collective goals. We considered a heterogeneous team comprised of two types of robots that make different decisions regarding this social dilemma. We proposed a decentralized approach for completing the task and studied the sustainability and efficiency of teams following that approach. We then proposed an institutional approach that provides tighter coordination for the team by giving one robot the role of institutional assembler. When following this approach, distinctions between robots are eliminated and the social dilemma loses its impact on the robot team. We studied how the institutional approach affects sustainability and efficiency, and what institutional costs are acceptable to take advantage of the improved coordination that is provided. In the latter part of this chapter we analyzed the robustness of the institutional approach to different environments, and we applied our modeling methodology in order to obtain *a priori* (with parameters calibrated from geometric considerations as opposed to data gathered from simulations) probabilistic macroscopic models of the system.

Chapter 9

Conclusions and Future Work

9.1 Conclusion

This dissertation presents several advances to institutional robotics, an approach to the coordination of distributed robotic systems that takes inspiration from the social sciences, in particular from institutional economics. This approach aims to provide a comprehensive strategy for specifying complex social interactions among a team of robots and possibly between a team of robots and human actors. The advances presented herein constitute an important stepping stone for this objective to be realized.

The benefits of such advances are threefold. First, the development of formal methods for control and modeling of distributed robotic systems under the institutional approach allows researchers to build upon our work when designing distributed robotic applications that require complex social interactions among robots. While the original set of guidelines for institutional robotics presented in [Silva and Lima, 2007] is a indispensable tool for conceiving the type of coordination strategies we are interested in, it leaves too many open possibilities for how researchers could actually apply them. This opens the door for *ad hoc* methods to be developed, possibly hampering the progress of the institutional robotics approach by not allowing distinct works to be compared or combined. Our work provides a baseline of methods that can be used "as is" or extended to further advance the study of the approach. By focusing on a modular strategy for our formal definition of institutions, we allow for the combination of work from different sources, while providing guarantees that relevant qualitative properties are maintained.

Second, the synthesis of the proposed control and modeling methods allows us to analyze behavior dynamics at the macroscopic level while grounding the models directly in the institutional controllers of the individual robots. While this idea was already present in the multi-level modeling methodology [Martinoli et al., 2004], we can now incorporate institutional rules capturing behaviors inspired in human societies. Also, the modularity of our approach, in terms of both controller design and model construction, allows for the combination of distinct controllers and models in a way that the original multi-level modeling methodology could not.

For instance, if we intend to design a high-level controller for an intelligent vehicle in a particular traffic scenario, we can start from the road code rules used in that particular situation to design an institutional agent controller that accomplishes the necessary navigation and coordination. With our approach we can analyze the scenario at the macroscopic level, using this controller as a starting point, possibly discovering new behavior dynamics for the human version of the same traffic scenario. The developed controller and model can then be combined with similar ones for other traffic scenarios, obtaining a more general controller capable of handling multiple scenarios and a more interesting model capturing more general dynamics of the problem.

Third, the novel implementation of the institutional robotics approach, using the developed formal methods and carried out over three different case studies, provides important validation that the approach is able to capture complex social interactions between robots. As expected, and observed in our results, considering such interactions improves the coordination of robotic teams and increases the performance of distributed robotic systems. Nevertheless, the institutional approach is not beneficial in all situations. For instance, we observed how in certain cases (e.g., when too many robots are dedicated exclusively to coordination) a self-organized solution might outperform our approach. This is not unexpected and reinforces our believe that the cost of using an institutional approach (in terms of available work force, energy costs, communication costs, etc.) must be carefully studied and taken into account when designing the coordination strategy.

The first objective for our work was to formalize institutional robotics' concepts from a computer science perspective, leading to coordination and control methods for distributed robotic systems where complex social interactions are taken into account. Our contributions and conclusions for this objective can be summarized as follows:

- We introduced Executable Petri Nets, an extension to the Petri net formalism. EPNs have associated actions and conditions that allow us to use them in order to specify behaviors that can be executed in robots. By guaranteeing the properties of safeness and liveness we can avoid problems during execution.

- We developed a methodology for distributed robotic systems based on the formalization of the concept of institution. This formalization takes institutions as coordination artifacts distributed over the robotic system and specified by EPNs and deontic operators. A composition algorithm guided by these operators allows for sets of institutions to be composed into an IAC that represents the institutional environment of each robot. Its replication in all nodes of a distributed robotics system provides the necessary coordination.

- We proved that this composition algorithm preserves the properties of safeness and liveness of the institutions.

The second objective for our work was to implement and validate concepts from institutional robotics, both in simulated and real robots, for laboratory scenarios designed to put forward relevant questions about the institutional approach, comparing the results against other existing approaches. Our contributions and conclusions for this objective can be summarized as follows:

- We validated the IAC methodology by implementing two versions of the wireless connected swarm case study. The original version of the case study uses an FSA controller. The second version uses an IAC composed of one individual behavior and two institutions designed to perform the task similarly. Submicroscopic model results show a good agreement between the two versions. We conclude that the IAC approach is able to replicate results produced with other control approaches.

- We obtained a real-world implementation of the wireless connected swarm case study, able to maintain the wireless connectivity of a swarm of (as many as) 40 real, resource-constrained robots. The submicroscopic model was validated by this real-world implementation by analyzing several different metrics (connectivity, dispersion and displacement).

- In the corridor case study we have shown that institutional roles can effectively help a distributed robotic system coordinate and improve performance in a given task. We also concluded that not all conditions are suitable to the application of an institutional approach, with self-organization being sufficient for small teams to obtain good performances. We have shown that it is possible through collective decision-making for robotics teams to adapt the parameters of institutions in order to adapt to dynamic environments.

- In the piece assembly case study we have shown that the introduction of institutions in distributed robotic tasks that involve a social dilemma can improve performance, efficiency and sustainability. We introduce sustainability as the ability of a robotic team to keep in members operational in tasks that deal with loss and gain of energy. We study how heterogeneity affects a system efficiency and sustainability, when heterogeneity is related to what priority robots give to individual vs collective goals. We conclude that an institutional approach can circumvent the problems presents by heterogeneity by enforcing a specific type of behavior. We also test the robustness of both the institutional approach and a purely decentralized approach. We conclude that while the coordination enforced by the institution improves performance it also hampers robustness.

The third objective for our work was to model, using formal mathematical methods, the distributed robotic systems designed and implemented under the institutional approach, providing sound tools for design evaluation and performance analysis and further comparison with other approaches. Our contributions and conclusions for this objective can be summarized as follows:

- We introduced an approach to the probabilistic modeling of distributed robotic systems controlled by IACs. Our approach follows a multi-level modeling methodology, focusing mainly on the macroscopic level and using the IAC as a starting point for a Generalized Stochastic Petri Net probabilistic model. We further extend the GSPN models with the introduction of an environmental information layer, where information not present in the IAC can be represented. The modular approach to controller and model design allows the designer to select relevant sections of the IAC, represent them as GSPN, and perform steady state analysis to obtain accurate prediction of the performance of the system.

- We applied our modeling approach to the wireless connected swarm case study and to the piece assembly case study. In the former case study, we were able to construct a GSPN model for the overall state distribution of the system. Using data gathered from the submicroscopic model in order to estimate the transition rates necessary for our GSPN model we were able to observe very good agreement between macroscopic and submicroscopic results. In the latter case study, we extended the GSPN constructed from the EPN of the institution with an environmental information layer. By estimating the transition rates from the physical properties of the environment we were able to obtain an *a priori* probabilistic model of the system that showed good agreement between macroscopic and microscopic results.

9.2 Future Work

As mentioned previously, the advances presented in this thesis provide a baseline of methods that can be used in the current state of development or extended by interested researchers in order the further advance the study of institutional robotics. There are ample opportunities to improve the formalization and implementation of concepts from the institutional approach, as well as improve the modeling of distributed robotic systems that follow it.

On the formalization of institution and the IAC methodology, we are interested in the following points:

- Improving the representation of deontic powers in the formalization of institutions would allow for a composition scheme that is not so restrictive in terms of possible concurrent execution of institutions.

- Distinct formalization of different institutional forms (roles, norms, organizations, hierarchies) would also improve the composition scheme while at the same time open interesting possibilities for the design of institutional agent controllers.

- The creation, adaptation, and eventual elimination, of institutions is one of the most interesting and necessary developments for the approach. Methods for collective decision-making can possibly aggregate an often-executed sequence of actions into an institution, allowing robots to abstract a complex social behavior into a formally represented and inspectable institution. The methods by which institutions are created are still not completely understood for human societies, but inspiration could be obtained in Elinor Ostrom's institutional analysis and development framework [Ostrom et al., 1994, Ostrom, 2005].

On the implementation and validation of institutional robotics' concepts, we are interested in the following points:

- Validation of the advances described in the previous set of point is clearly a necessity. The development of case studies where the creation (and elimination) of institutions can be studied is not a trivial exercise.

- The implementation of case studies (or particular situations) where institutions are shared between robots and the IAC must be altered during execution should also be considered.

- The next main objective in the implementation and validation of the institutional robotics approach must be to consider experiments in real-world scenarios populated with human actors. Robots should interact with these actors and such interaction should not be based on spoken dialogue (at least in an initial approach) but rather on coordination through common rules, described as institutions. Such experiments would validate our intuition that by considering complex social interactions in distributed robotic systems we will ease the effort of their transition to real-world environments populated with human actors.

On the probabilistic modeling of distributed robotic systems under the institutional approach, we are interested in the following points:

- Following the intuition from the item above, the modeling of mixed human-robot distributed systems should be considered. If both human actors and robots share institutions who can be represented through EPN, it can be assumed that similar GSPN macroscopic models can capture the dynamics of such systems.

- Macroscopic modeling of the corridor case study was not performed up to this point. Applying our modeling approach using the IAC as starting point, as well as obtaining a macroscopic model from the designed FSA, would allows to compare different approaches to modeling.
- Studying how to use the obtained macroscopic models for analysis and optimization of control parameters.

It would also be interesting to discuss with researchers from the areas of inspiration of institutional robotics, especially economists interested in multi-agent simulation. It is possible that the formalisms and models developed for distributed robotic systems can be used as a tool for such researchers to obtain better models of human economic behavior. If this is the case, it would close the loop between institutional economics and institutional robotics, yielding a truly interdisciplinary research effort.

Appendix A

Institutional Agent Controller Composition Proof

In this appendix we present the proof for Proposition 1 presented in Section 4.6.

Proof We assume that Ind and $Inst_i$ ($i = 1, \ldots, n$) verify the properties of safeness and liveness described in Definitions 3 and 4. We follow the IAC construction steps from Algorithm 2, verifying at each point if such properties are preserved.

- Line 3: adding the EPN Ind to the empty EPN IAC_{net} trivially preserves the properties since Ind is safe and live.
- Lines 4-6: At this point, IAC_{net} is safe and live. We need to show that adding $Inst_i$ to IAC_{net} still produces a safe and live net. Let $IAC_{net} = (P, T, A, X)$ and $Inst_i = (P_{Inst,i}, T_{Inst,i}, A_{Inst,i}, X_{Inst,i})$. We remove the weight function w from the tuples since we have specified that all transitions in EPNs have weight one. We designate as $IAC_{net} \cup Inst_i$ the inclusion of $Inst_i$ in IAC_{net} and define it as

$$IAC_{net} \cup Inst_i = (P \cup P_{Inst,i}, T \cup T_{Inst,i}, A \cup A_{Inst,i}, [X\ X_{Inst,i}]), \tag{A.1}$$

  where $[X\ X_{Inst,i}]$ is the concatenation of states $X = [x(p_1), \ldots, x(p_m)]$ and $X_I = [x_{Inst,i}(p_1), \ldots, x_{Inst,i}(p_k)]$, and $m = |P|$, $k = |P_{Inst,i}|$. Since there is no arc between places and transitions in IAC_{net} and places and transitions in $Inst_i$ each reachable state of $IAC_{net} \cup Inst_i$ will be a combination of the reachable states of IAC_{net} and $Inst_i$:

$$R[IAC_{net} \cup Inst_i] = \{[X\ X_{Inst,i}] : X \in R[IAC_{net}], X_{Inst,i} \in R[Inst_i]\}. \tag{A.2}$$

  - Safeness: for all states $Y = [X\ X_{Inst,i}] \in R[IAC_{net} \cup Inst_i]$, $p \in P \cup P_{Inst,i}$, we have $y(p) \leq 1$ since, if $p \in P$, then $y(p) = x(p) \leq 1$, and, if $p \in P_{Inst,i}$, then $y(p) = x_{Inst,i}(p) \leq 1$.

  - Liveness: for any transition $t \in T \cup T_{Inst,i}$, if $t \in T$, then there exists a sample path $s \in T^*$ such that t can fire from any state $X \in R[IAC_{net}]$. The same argument is valid if $t \in T_{Inst,i}$. Thus, for any transition $t \in T \cup T_{Inst,i}$ there exists a sample path $s \in (T \cup T_{Inst,i})^*$ such that t can fire from any state $X \in R[IAC_{net} \cup Inst_i]$, since $T^*, T^*_{Inst,i} \subseteq (T \cup T_{Inst,i})^*$.

- Lines 8-12: we add a macro place m_{Ind} to represent Ind in the higher layer and bidirectional arcs from m_{Ind} to each transition $t \in T_{Ind}$, where $T_{Ind} \subset T$ and $IAC_{net} = (P, T, A, X)$. The resulting net is defined as:

$$
\begin{aligned}
IAC_{net} = (&P \cup \{m_{Ind}\}, \\
&T, \\
&A \cup \{(m_{Ind}, t_1), (t_1, m_{Ind}), \ldots, (m_{Ind}, t_k), (t_k, m_{Ind})\}, \\
&[X\ 1]),
\end{aligned}
\tag{A.3}
$$

where $k = |T_{Ind}|$, $\{t_1, \ldots, t_k\} = T_{Ind}$, and $[X\ 1]$ is the concatenation of the state of the net with the marking of place m_{Ind}.

  - Safeness: all arcs added are bidirectional, meaning that when any transition $t \in T_{ind}$ fires, the marking of place m_{Ind} remains unaltered. Since no other arcs are added between transitions and places, all other places are safe. Thus, the resulting IAC_{net} is safe.

  - Liveness: the added arcs (provided that m_{Ind} is marked) have no effect on the enabling of transitions in T_{Ind} or the state changes caused by their firing. Thus, liveness is preserved.

- Lines 13-22: similarly to the previous point, we add a macro place m_{Ii} to represent each institution $Inst_i$ in the higher layer and bidirectional arcs from m_{Ii} to each transition $t \in T_{Inst,i}$, where $T_{Inst,i} \subset T$ and $IAC_{net} = (P, T, A, X)$. In the case of institutions we also add a control structure composed of an idle place $idle_{Ii}$ and transitions $t_{initial,i}$ and $t_{final,i}$ associated with the initial and final conditions $initial_{I,i}$ and $final_{I,i}$, respectively, and connections as shown in Fig. 4.5. The resulting net is defined as:

$$
\begin{aligned}
IAC_{net} = (&P \cup \{m_{Ii}, idle_{Ii}\}, \\
&T \cup \{t_{initial,i}, t_{final,i}\}, \\
&A \cup \{(m_{Ii}, t_1), (t_1, m_{Ii}), \ldots, (m_{Ii}, t_k), (t_k, m_{Ii}), \\
&\qquad (idle_{Ii}, t_{initial,i}), (t_{initial,i}, m_{Ii}), (m_{Ii}, t_{final,i}), (t_{final,i}, idle_{Ii})\}, \\
&[X\ 0\ 1]),
\end{aligned}
\tag{A.4}
$$

where $k = |T_{Inst,i}|$, $\{t_1, \ldots, t_k\} = T_{Inst,i}$, and $[X\ 0\ 1]$ is the concatenation of the state of the net with the marking of places m_{Ii} and $idle_{Ii}$ in this sequence.

– Safeness: considering only the macro place m_{Ii} and its connections to the transitions in $T_{Inst,i}$ we are in the same situation as previously (with m_{Ind}). In order for IAC_{net} to be safe m_{Ii} and $idle_{Ii}$ also need to be safe. The added control structure moves single tokens between these two places through transitions $t_{initial,i}$ and $t_{final,i}$. The only reachable (sub-)states for these places are [0 1] and [1 0], thus they are safe and so is IAC_{net}.

– Liveness: as before, provided that m_{Ii} is marked, the added arcs have no effect on the enabling of transitions in $T_{Inst,i}$ or the state changes caused by their firing. If m_{Ii} is not marked, then $idle_{Ii}$ is marked and $t_{initial,i}$ is enabled. So it is sufficient to append the firing of transition $t_{initial,i}$ to any sample path $s \in T^*$ that would allow a transition $t \in T$ to fire. The transitions $t_{initial,i}$ and $t_{final,i}$ are clearly live.

We are left to prove that adding the control structures structures that regulate the possible concurrent execution of behaviors preserves the properties of interest. These structures are added only to the higher layer of the net and are dictated by the deontic operators of institutions. We assume that IAC_{net} verifies the properties and that adding controls structures related to institution I_i, dependent on the deontic operator $d_{I,i}$, preserves those properties.

- Lines 25-27, deontic operator *AllowAll*: in the case of this operator no further structures are added, so trivially IAC_{net} retains the properties.

- Lines 28-32, deontic operator *StopInd*: in the case of this operator we add a control structure composed of a new place $idle_{Ind,Ii}$ and transitions $stop\ execution_{Ind,Ii}$ and $restart\ execution_{Ind,Ii}$, as shown in Fig. 4.5 (transition labels are not shown), to $IAC_{net} = (P, T, A, X)$. The resulting net is defined as:

$$
\begin{aligned}
IAC_{net} = (&P \cup \{idle_{Ind,Ii}\}, &\text{(A.5)}\\
&T \cup \{stop\ execution_{Ind,Ii}, restart\ execution_{Ind,Ii}\},\\
&A \cup \{(m_{Ind}, stop\ execution_{Ind,Ii}), (m_{Ii}, stop\ execution_{Ind,Ii}),\\
&\quad (stop\ execution_{Ind,Ii}, idle_{Ind,Ii}), (stop\ execution_{Ind,Ii}, m_{Ii}),\\
&\quad (idle_{Ind,Ii}, restart\ execution_{Ind,Ii}), (idle_{Ii}, restart\ execution_{Ind,Ii}),\\
&\quad (restart\ execution_{Ind,Ii}, m_{Ii}), (restart\ execution_{Ind,Ii}, idle_{Ii}), \},\\
&[X\ 0]), &\text{(A.6)}
\end{aligned}
$$

where $[X\ 0]$ is the concatenation of the state of the net with the marking of place $idle_{Ind,Ii}$.

– Safeness: the added transitions change only the marking of places m_{Ind} and $idle_{Ind,Ii}$, given that arcs to other places are bidirectional. Since we have restricted arc weights to be equal to one, it is easily observed that the added transitions will always preserve the safeness of those places.

- Liveness: as before, provided that m_{Ind} is marked, the added arcs have no effect on the enabling of transitions in T or the state changes caused by their firing. If m_{Ind} is not marked, one of two possible scenarios are true: i) there is a sample path $s \in T^*$ such that after all transition in s have fired, m_{Ind} is marked (because the IAC_{net} before the addition of the new control structure is live); ii) $idle_{Ind,Ii}$ is marked. If the first scenario is true there is also a sample path $s' \in T^*$ such that after its firing m_{Ii} is marked. In this case transition $stop\ execution_{Ind,Ii}$ is enabled and can fire, resulting in $idle_{Ind,Ii}$ becoming marked if it does. If $idle_{Ind,Ii}$ is marked then there is a sample path $s'' \in T^*$ such that after its firing $idle_{Ii}$ is marked. In this case transition $restart\ execution_{Ind,Ii}$ is enabled and can fire, resulting in m_{Ind} being marked if it does. Thus, the new IAC_{net} is live.

- Lines 33-39, deontic operator *StopInst*: in the case of this operator we add a control structure similar to the one added for *StopInd*. The same arguments apply for the preservation of safeness and liveness.

- Lines 28-32, 33-39, deontic operator *StopAll*: the structures added for this operator are a combination of those added for *StopInd* and *StopInst*. Again, safeness and liveness are preserved.

Thus, the composition of individual behavior, institutions, their respective macro and idle places, and structures added to regulate their concurrent execution, preserves the properties of safeness and liveness, resulting in an IAC that verified those properties.

□

Appendix B

Piece Assembly Case Study Extra Material

Algorithm 3 computes the prospect vector P for all slots of the assembly site used by robots following the decentralized approach in the piece assembly case study.

Algorithm 3 Prospect: Function to compute the prospect vector (P) for the assembly site. For each slot i, P_i indicates if placement of component F in slot i would allow piece completion in the future. $P_i = 1$ if piece completion possible, else $P_i = 0$

Parameters:

C Number of different types of components at the building site;

S Number of slots at the assembly site;

C_i Component at slot i;

F Component collected by the robot

{Evaluation of prospect at each of the slots}
for $slot = S \rightarrow 1$ **do**
    $P_{slot} \leftarrow 0$
    if $C_{slot} \neq \phi$ **then**
        {Current slot is not available. We set its prospect to 0 and move onto next slot}
        Continue to next *slot*
    end if
        {We now check if succeeding components can be placed to complete the piece}
    $currentslot \leftarrow slot - 1$
    for $succeedingcomponent = F + 1 \rightarrow C$ **do**
        if $!(C_{currentslot} = \phi \vee C_{currentslot} = succeedingcomponent)$ **then**
            {An incorrect component is present. The piece cannot be completed.}
            Continue to next *slot*
        end if
        $currentslot \leftarrow currentslot - 1$
        if $currentslot < 1 \vee currentslot > S$ **then**
            {We have run out of slots at the assembly site. The piece cannot be completed}
            Continue to next *slot*
        end if
    end for
        {We now check if preceding components can be placed to complete the piece}
    $currentslot \leftarrow slot + 1$
    for $precedingcomponent = F - 1 \rightarrow 1$ **do**
        if $!(C_{currentslot} = \phi \vee C_{currentslot} = precedingcomponent)$ **then**
            {An incorrect component is present. The piece cannot be completed.}
            Continue to next *slot*
        end if
        $currentslot \leftarrow currentslot + 1$

```
    if currentslot < 1 ∨ currentslot > S then
      {We have run out of slots at the assembly site. The piece cannot be completed}
      Continue to next slot
    end if
  end for
  P_slot ← 1
end for
return P
```

Bibliography

[Ajmone Marsan et al., 1989] Ajmone Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G., and Cumani, A. (1989). The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15(7):832–846.

[Ajmone Marsan et al., 1995] Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1995). *Modelling with generalized stochastic Petri nets*. John Wiley & Sons, New York.

[Ajmone Marsan et al., 1984] Ajmone Marsan, M., Conte, G., and Balbo, G. (1984). A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122.

[Aldewereld et al., 2010] Aldewereld, H., Álvarez Napagao, S., Dignum, F., and Vázquez-Salceda, J. (2010). Making Norms Concrete. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems - AAMAS 2010*, pages 807–814.

[Alonso, 2004] Alonso, E. (2004). Rights and Argumentation in Open Multi-Agent Systems. *Artificial Intelligence Review*, 21:3–24.

[Anderies et al., 2004] Anderies, J. M., Janssen, M. A., and Ostrom, E. (2004). A Framework to Analyze the Robustness of Social-ecological Systems from an Institutional Perspective. *Ecology And Society*, 9(1).

[Bause and Kritzinger, 2002] Bause, F. and Kritzinger, P. S. (2002). *Stochastic Petri Nets - An Introduction to the Theory*. Friedr. Vieweg &Sohn Verlag, Braunschweig/Wiesbaden (Germany), second edition.

[Beckers et al., 1994] Beckers, R., Holland, O. E., and Deneubourg, J.-L. (1994). From local actions to global tasks: stigmergy and collective robotics. In Brooks, R. and Maes, P., editors, *Proceedings of ALIFE IV*, page 189. MIT Press, Cambridge.

[Beni, 2005] Beni, G. (2005). From Swarm Intelligence to Swarm Robotics. In Sahin, E. and Spears, W. M., editors, *Swarm Robotics*, pages 1–9. Springer.

[Berman et al., 2011] Berman, S., Halasz, A., and Hsieh, M. A. (2011). Ant-Inspired Allocation: Top-Down Controller Design for Distributing A Robot Swarm among Multiple Tasks. In Xiao, Y., editor, *Bio-inspired Computing and Networking*. CRC Press, New York.

[Berman et al., 2009] Berman, S., Halasz, A., Hsieh, M. A., and Kumar, V. (2009). Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937.

[Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

[Bonet et al., 2007] Bonet, P., Lladó, C., Puijaner, R., and Knottenbelt, W. (2007). PIPE v2.5: A Petri net tool for performance modelling. In *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*, San Jose, Costa Rica.

[Bosch-Domènech and Sunder, 2000] Bosch-Domènech, A. and Sunder, S. (2000). Tracking the Invisible Hand: Convergence of Double Auctions to Competitive Equilibrium. *Computational Economics*, 16(3):257–284.

[Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. MIT Press, Cambridge, MA.

[Brito et al., 2013] Brito, M. D., Hübner, J. F., and Bordini, R. H. (2013). Programming Institutional Facts in Multi-Agent Systems. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII, Lecture Notes in Computer Science Volume 7756*, pages 158–173. Springer.

[Caldas, 2001] Caldas, J. M. C. (2001). *Escolha e instituições: análise económica e simulação multiagentes (Choice and institutions: economical analysis and multi-agent simulation)*. Celta Editora.

[Cao et al., 1997] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B. (1997). Cooperative Mobile Robotics : Antecedents and Directions. *Autonomous robots*, 23:1–23.

[Caprari and Siegwart, 2005] Caprari, G. and Siegwart, R. (2005). Mobile micro-robots ready to use: Alice. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, volume 1, pages 3295–3300.

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

[Christensen, 2005] Christensen, A. L. (2005). Efficient Neuro-Evolution of Hole-Avoidance and Phototaxis - TR/IRIDIA/2005-14,. Technical report, IRIDIA, Universite Libre de Bruxelles, Bruxelles, Belgium.

[Cianci et al., 2006] Cianci, C., Raemy, X., Pugh, J., and Martinoli, A. (2006). Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Simulation of Adaptive Behaviour (SAB-2006), Swarm Robotics Workshop*, pages 103–115, Rome, Italy.

[Coase, 1992] Coase, R. H. (1992). The Institutional Structure of Production. *The American Economic Review*, 82(4):713–719.

[Conte and Castelfranchi, 1995] Conte, R. and Castelfranchi, C. (1995). *Cognitive and social action*. Londra: London University College of London Press.

[Correll and Martinoli, 2011] Correll, N. and Martinoli, A. (2011). Modeling and designing self-organized aggregation in a swarm of miniature robots. *International Journal of Robotics Research*, 30(5):615–626.

[Costelha and Lima, 2012] Costelha, H. and Lima, P. U. (2012). Robot task plan representation by Petri nets: modelling, identification, analysis and execution. *Autonomous Robots*, 33(4):337–360.

[Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory, Second Edition*. Wiley, New York.

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

[Cuni et al., 2004] Cuni, G., Esteva, M., Garcia, P., Puertas, E., Sierra, C., and Solchaga, T. (2004). MASFIT : Multi-Agent System for FIsh Trading. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 710–714, València, Spain.

[David and Alla, 2010] David, R. and Alla, H. (2010). *Discrete, Continuous, and Hybrid Petri Nets*. Springer-Verlag, Berlin/Heidelberg.

[Dias et al., 2006] 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.

[Dias et al., 2004] Dias, M. B., Zlot, R., Zinck, M., Gonzalez, J. P., and Stentz, A. T. (2004). A Versatile Implementation of the TraderBots Approach for Multirobot Coordination. In Groen, F. C. A., Bonarini, A., and Amato, N., editors, *Intelligent Autonomous Systems 8 (Proceedings of the International Conference on Intelligent Autonomous Systems IAS 2004)*, pages 325–334. IOS Press.

[Dietz et al., 2003] Dietz, T., Ostrom, E., and Stern, P. C. (2003). The struggle to govern the commons. *Science (New York, N.Y.)*, 302(5652):1907–12.

[Dingle et al., 2009] Dingle, N., Knottenbelt, W., and Suto, T. (2009). PIPE2: a tool for the performance evaluation of generalised stochastic Petri Nets. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):34–39.

[Dorigo et al., 2012] Dorigo, M., Floreano, D., Gambardella, L., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., Burnier, D., Campo, A., Christensen, A., Decugnière, A., Caro, G. D., Ducatelle, F., Ferrante, E., Förster, A., Guzzi, J., Longchamp, V., Magnenat, S., Gonzalez, J. M., Mathews, N., de Oca, M. M., O'Grady, R., Pinciroli, C., Pini, G., Rétornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stuetzle, T., Trianni, V., Tuci, E., Turgut, A., and Vaussard, F. (2012). Swarmanoid : a novel concept for the study of heterogeneous robotic swarms IRIDIA ,Äì Technical Report Series Technical Report No . *IEEE Robotics & Automation Magazine*, (July):(in press).

[Ducatelle et al., 2010] Ducatelle, F., Caro, G. A. D., and Gambardella, L. M. (2010). Cooperative Self-Organization in a Heterogeneous Swarm Robotic System. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 87–94.

[Durfee, 2004] Durfee, E. H. (2004). Challenges to Scaling-Up Agent Coordination Strategies. In Wagner, T. A., editor, *An Application Science for Multi-Agent Systems*, volume 34, pages 113–132. Springer.

[Esparza, 1994] Esparza, J. (1994). Model checking using net unfoldings. *Science of Computer Programming*, 23:151–195.

[Esteva and Padget, 2000] Esteva, M. and Padget, J. (2000). Auctions without Auctioneers : Distributed Auction Protocols. In *Agent Mediated Electronic Commerce II Lecture Notes in Computer Science Volume 1788*, volume 27957, pages 220–238. Springer.

[Esteva et al., 2002] Esteva, M., Padget, J., and Sierra, C. (2002). Formalizing a language for institutions and norms. *Intelligent Agents VIII*, pages 348–366.

[Esteva et al., 2000] Esteva, M., Rodriguez-Aguilar, J., Arcos, J., Sierra, C., and Garcia, P. (2000). Institutionalising open multi-agent systems. In *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, pages 381–382. IEEE.

[Esteva et al., 2001] Esteva, M., Rodríguez-Aguilar, J.-A., Sierra, C., Garcia, P., and Arcos-Rosell, J.-L. (2001). On the Formal Specification of Electronic Institutions. In *Agent Mediated Electronic Commerce Lecture Notes in Computer Science Volume 1991*, pages 126–147. Springer.

[Esteva et al., 2004] Esteva, M., Rosell, B., Rodriguez-Aguilar, J. A., and Arcos-Rosell, J.-L. (2004). AMELI : An Agent-based Middleware for Electronic Institutions. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 236–243, New York, USA.

[Evans et al., 2010] Evans, W. C., Mermoud, G., and Martinoli, A. (2010). Comparing and modeling distributed control strategies for miniature self-assembling robots. In *International Conference on Robotics and Automation*, pages 1438–1445, Anchorage, Alaska, USA.

[Gerkey and Mataric, 2002] Gerkey, B. P. and Mataric, M. J. (2002). Sold !: Auction Methods for Multirobot Coordination. *IEEE Transactions on Robotics*, 18(5):758–768.

[German and Lindemann, 1994] German, R. and Lindemann, C. (1994). Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation*, 20(1-3):317–335.

[Girault and Valk, 2003] Girault, C. and Valk, R. (2003). *Petri Nets for Systems Engineering*. Springer.

[Gode and Spear, 2004] Gode, D. K. and Spear, S. (2004). Convergence of Double Auctions to Pareto Optimal Allocations in the Edgeworth Box.

[Gode and Sunder, 1993] Gode, D. K. and Sunder, S. (1993). Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. 101(1):119–137.

[Gross et al., 2006] Gross, R., Bonani, M., Mondada, F., and Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130.

[Guo et al., 2011] Guo, H., Meng, Y., and Jin, Y. (2011). Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm. In *International Conference on Robotics and Automation*, pages 3205–3210, Shangai, China.

[Hahn et al., 2007] Hahn, C., Fley, B., Florian, M., Spresny, D., and Fischer, K. (2007). Social reputation: A mechanism for flexible self-regulation of multiagent systems. *Journal of Artificial Societies and Social Simulation*, 10(1).

[Hamilton, 1919] Hamilton, W. H. (1919). The institutional approach to economic theory. *The American Economic Review*, 9(1):309–318.

[Hardin, 1968] Hardin, G. (1968). The tradegy of the commons. *Science*, (June).

[He and Lemmon, 2000] He, K. X. and Lemmon, M. D. (2000). Liveness Verification of Discrete Event Systems Modeled by n-Safe Ordinary Petri Nets. *Application and Theory of Petri Nets 2000 - LNCS*, 1825:227–243.

[Hexmoor et al., 2006] Hexmoor, H., Venkata, S. G., and Hayes, D. (2006). Modelling social norms in multiagent systems. *Journal of Experimental & Theoretical Artificial Intelligence*, 18(1):49–71.

[Hodgson, 1988] Hodgson, G. M. (1988). *Economics and Institutions: A Manifesto for a Modern Institutional Economics*. Polity Press, Cambridge.

[Hodgson, 1998] Hodgson, G. M. (1998). The Approach of Institutional Economics. *Journal of Economic Literature*, XXXVI(March):166–192.

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

[Hodgson, 2006] Hodgson, G. M. (2006). What are Institutions? *Journal of Economic Issues*, XL(1):1–25.

[Hoff et al., 2011] Hoff, N., Wood, R., and Nagpal, R. (2011). Effect of sensor and actuator quality on robot swarm algorithm performance. In *International Conference on Intelligent Robots and Systems*, pages 4989–4994, San Francisco, CA, USA.

[Kelling, 1995] Kelling, C. (1995). TimeNET-Sim - a Parallel Simulator for Stochastic Petri Nets. In *Proceedings of the 28th Annual Simulation Symposium*, volume 29, pages 250–258, Phoenix, AZ, USA.

[Krieger and Billeter, 2000] Krieger, M. J. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84.

[Levis et al., 2004] Levis, P., Madden, S., Polastre, J., Szewczyk, R., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D. (2004). Tinyos: An operating system for sensor networks. In *Ambient Intelligence*. Springer Verlag.

[Li et al., 2004] Li, L., Martinoli, A., and Abu-Mostafa, Y. S. (2004). Learning and Measuring Specialization in Collaborative Swarm Systems. *Adaptive Behavior*, 12(3-4):199–212.

[Lochmatter et al., 2008] Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., and Martinoli, A. (2008). SwisTrack - a flexible open source tracking software for multi-agent systems. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4004–4010. Ieee.

[Malsch and Weiß, 2001] Malsch, T. and Weiß, G. (2001). Conflicts in Social Theory and Multiagent Systems: on Importing Sociological Insights into Distributed AI. In Tessier, C., Chaudron, L., and Müller, H.-J., editors, *Conflicting agents: conflict management in multi-agent systems*, pages 111–149. Kluwer Academic Publishers.

[Martinoli et al., 2004] Martinoli, A., Easton, K., and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436.

[Menard and Shirley, 2005] Menard, C. and Shirley, M. M. (2005). Introduction. In Menard, C. and Shirley, M. M., editors, *Handbook of New Institutional Economics*, pages 1–18. Springer.

[Mermoud et al., 2014] Mermoud, G., Upadhyay, U., Evans, W. C., and Martinoli, A. (2014). Top-Down vs Bottom-Up Model-Based Methodologies for Distributed Control: A Comparative Experimental Study. In *Proc. of the Twelfth Int. Symp. Experimental Robotics, December 2010*, New Delhi, India. Springer Tracts in Advanced Robotics, Vol. 79. To appear.

[Michel, 2004] Michel, O. (2004). Webots TM : Professional Mobile Robot Simulation. *Advanced Robotic*, 1(1):39–42.

[Mondada et al., 2009] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65.

[Mondada et al., 2005] Mondada, F., Gambardella, L. M., Floreano, D., Nolfi, S., Deneubourg, J.-L., and Dorigo, M. (2005). The cooperation of swarm-bots: physical interactions in collective robotics. *Robotics & Automation Magazine, IEEE*, 12(2):21–28.

[Mondada et al., 2004] Mondada, F., Pettinaro, G., and Guignard, A. (2004). SWARM-BOT: A new distributed robotic concept. *...Robots*, 17(2/3):193–221.

[Murata, 1989] Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580.

[Nembrini et al., 2002] Nembrini, J., Winfield, A. F. T., and Melhuish, C. (2002). Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots. *From animals to animats 7: Proceedings of the seventh international conference on simulation of adaptive behavior*, pages 273–282.

[North, 1993] North, D. C. (1993). The New Institutional Economics and Development. *EconWPA Economic History*, (January).

[Omicini et al., 2004] 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, 1990] Ostrom, E. (1990). *Governing the commons: the evolution of institutions for collective action*. Cambridge University Press.

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

[Ostrom et al., 1994] Ostrom, E., Gardner, R., and Walker, J. (1994). *Rules, Games, & Common-Pool Resources*. University of Michigan Press.

[Parker, 1998] Parker, L. E. (1998). ALLIANCE : An Architecture for Fault Tolerant Multirobot Cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240.

[Payton et al., 2001] Payton, D., Daily, M., Estowski, R., Howard, M., and Lee, C. (2001). Pheromone robotics. *Autonomous Robots*, 11:319–324.

[Pereira et al., 2010] Pereira, J. N., Christensen, A. L., Silva, P., and Lima, P. U. (2010). Coordination Through Institutional Roles in Robot Collectives (extended abstract). 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.

[Pereira et al., 2011] Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2011). Formalizing Institutions as Executable Petri Nets for Distributed Robotic Systems. In Lenaerts, T., Giacobini, M., Bersini, H., Bourgine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011*, pages 646–653, Paris, France. MIT Press.

[Pereira et al., 2013] Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2013). An Experimental Study in Wireless Connectivity Maintenance Using up to 40 Robots Coordinated by an Institutional Robotics Approach (accepted). In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[Pereira et al., 2014a] Pereira, J. N., Silva, P., Lima, P. U., and Martinoli, A. (2014a). Formalization, Implementation, and Modeling of Institutional Controllers for Distributed Robotic Systems (in print). *Artificial Life*, 20(1).

[Pereira et al., 2014b] Pereira, J. N., Tarapore, D., Silva, P., Martinoli, A., and Lima, P. U. (2014b). Considering Institutions in Unsustainable Robotic Systems (in preparation). *Autonomous Agents and Multi-Agent Systems*.

[Petri, 1966] Petri, C. A. (1966). *Kommunikation mit Automaten*. New York, second edi edition.

[Pettit, 2003] Pettit, P. (2003). Groups with Minds of Their Own. In Schmitt, F. F., editor, *Socializing Metaphysics*, pages 167–194. Rowman & Littlefield.

[Poteete and Ostrom, 2004] Poteete, A. R. and Ostrom, E. (2004). Heterogeneity , Group Size and Collective Action : The Role of Institutions in Forest Management. *Development and Change*, 35(3):435–461.

[Ren et al., 2005] Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *American Control Conference, Proceedings of the 2005*, pages 1859–1864.

[Ricci et al., 2005] 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.

[Sabater and Sierra, 2005] Sabater, J. and Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60.

[Sahin, 2005] Sahin, E. (2005). Swarm Robotics: From Sources of Inspiration to Domains of Application. In Sahin, E. and Spears, W. M., editors, *Swarm Robotics*, pages 10–20. Springer.

[Santos and Chaimowicz, 2011] Santos, V. G. and Chaimowicz, L. (2011). Hierarchical congestion control for robotic swarms. In *International Conference on Intelligent Robots and Systems*, pages 4372–4377, San Francisco, CA, USA.

[Searle, 1965] Searle, J. R. (1965). What is a speech act? In Stainton, R. J., editor, *Perspectives in the philosophy of language: a concise anthology*, pages 253–268. Broadview Press, 2000 edition.

[Searle, 2005] Searle, J. R. (2005). What is an institution? *Journal of Institutional Economics*, 1(1):1–22.

[Searle, 2006] Searle, J. R. (2006). Social Ontology : Some Basic Principles. *Anthropological Theory*, 6(March 2006):12–29.

[Senft et al., 2012] Senft, E., Pereira, J. N., and Martinoli, A. (2012). Consensus Problem in Robot Teams (DISAL-SP40). Technical report, DISAL, EPFL, Lausanne, Switzerland.

[Sierra et al., 2004] Sierra, C., Rodriguez-Aguilar, J. A., Noriega, P., Arcos-Rosell, J.-L., and Esteva, M. (2004). Engineering multi-agent systems as electronic institutions. *UPGRADE The European Journal for the Informatics Professional*, V(4):33–39.

[Silva, 2007] Silva, P. (2007). *Robótica Institucionalista: as Ciências do Artificial como Ciências do Humano (Institutional Robotics: Sciences of the Artificial as Human Sciences)*. Phd dissertation, Faculdade de Letras, Universidade de Lisboa.

[Silva and Lima, 2007] 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 et al., 2013] Silva, P., Pereira, J. N., and Lima, P. U. (2013). Institutional Robotics. Institutions for social robots. (submitted). *International Journal of Social Robotics.*

[Silva et al., 2008] Silva, P., Ventura, R., and Lima, P. U. (2008). Institutional environments. In *Proc. of Workshop AT2AI-6: From agent theory to agent implementation, AAMAS 2008 - 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 157–164, Estoril, Portugal.

[Tumer et al., 2002] Tumer, K., Agogino, A. K., and Wolpert, D. H. (2002). Learning sequences of actions in collectives of autonomous agents. In *Proc. of the 1st Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 378–385, Bologna, Italy.

[Tummolini and Castelfranchi, 2006] 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.

[Vail and Veloso, 2003] Vail, D. and Veloso, M. (2003). Dynamic Multi-Robot Coordination. In Schultz, A. C., Parker, L. E., and Schneider, F. E., editors, *Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II (Proceedings from the 2003 International Workshop on Multi-Robot Systems)*, pages 87–100. Springer.

[Varga, 2002] Varga, A. (2002). Software tools for networking: OMNeT++. *IEEE Network Interactive*, 16(4).

[Weyns et al., 2005a] Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T., and Ferber, J. (2005a). Environments for Multiagent Systems State-of-the-Art and Research Challenges. In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems*, pages 1–47. Springer, lecture no edition.

[Weyns et al., 2005b] Weyns, D., Schumacher, M., Ricci, A., Viroli, M., and Holvoet, T. (2005b). Environment , a first-order abstraction in multiagent systems. *Knowledge Engineering Review*, 20(2):127–141.

[Winfield et al., 2008] 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.

[Winfield and Nembrini, 2006] Winfield, A. F. T. and Nembrini, J. (2006). Safety in numbers : fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30–37.

[Zimmermann and Freiheit, 2000] Zimmermann, A. and Freiheit, J. (2000). Petri net modelling and performability evaluation with TimeNET 3.0. In *Computer Performance Evaluation.Modelling Techniques and Tools Lecture Notes in Computer Science Volume 1786*, pages 188–202.

[Zimmermann and Knoke, 2006] Zimmermann, A. and Knoke, M. (2006). Towards version 4.0 of TimeNET. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2006 13th GI/ITG Conference*, Nuremberg, Germany.

[Ziparo et al., 2011] Ziparo, V. A., Iocchi, L., Lima, P. U., Nardi, D., and Palamara, P. F. (2011). Petri Net Plans: A Framework for Collaboration and Coordination in Multi-Robot Systems. *Autonomous Agents and Multi-Agent Systems*, 23(3):344–383.