Reorganization in Dynamic Agent Societies

PhD Thesis

Juan Miguel Alberola Oltra

Supervisors: Dr. Ana García Fornes
Dr. Vicente Julián Inglada

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n. 46022, València. Spain

January 2013

Reorganization in Dynamic Agent Societies

A thesis submitted in partial fulfilment for the degree of:
Doctor of Philosophy in Computer Science
(Doctorado de Reconocimiento de Formas e Inteligencia Artificial)

Juan Miguel Alberola Oltra

Supervisors: Dr. Ana García Fornes
Dr. Vicente Julián Inglada

Grup de Tecnologia Informàtica i Intel·ligència Artificial

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n. 46022, València. Spain

January 2013

Als que ja no esteu amb nosaltres.

Nothing is constant, except change.

Joey Tempest.

Agraïments
Acknowledgments

M'agradaria dedicar unes línies a agrair a tots aquells que d'alguna forma o altra m'han ajudat i suportat durant tot este temps. Estos agraïments són per als que han estat al meu costat tant professionalment com personalment, però especialment m'agradaria nomenar a alguns d'ells.
Primerament m'agradaria nomenar als meus directors, Ana i Vicente Julián. Ells m'han donat moltes idees, suport i ànims que m'han ajudat en esta trajectòria, i sobretot la realimentació necessària de tot el que anava fent, fins i tot d'aquells articles que semblaven somnolents. Sense la seua paciència, ajuda i consells, esta tesi no haguera sigut possible. També m'agradaria mencionar especialment a Vicent Botti, amb qui he treballat i sobretot qui em va donar l'oportunitat de treballar al GTI, i a Agustín, pel seu assessorament durant la meua primera etapa al grup.
També m'agradaria agrair als meus companys de laboratori, dinars i articles el suport durant este temps i sobretot, les dosis de risses i humor, que tant necessàries són constantment: Toni, Maria, Elena, Jaume, Joan, Victor, Jose, Luis, Bexy, Mario, Raul i Ricard, a qui dec part de la implementació i integració en la plataforma.
I would like to also thank Tim for hosting me at Aberdeen and for his great support and suggestions during and after my stay and also as a mentor in the doctoral consortium. I would like to extend these thanks to Murat, Hengfei and Wamberto, for their help during my stay and make it so nice.
I per acabar, sobretot m'agradaria agrair als meus pares i a Ester el seu suport, ajuda i afecte durant estos anys. Ells són els que estan el dia a dia al meu costat i els que m'han recolzat i animat sempre.
A tots vosaltres, gràcies!!

Abstract

The new age of information technologies demands systems aimed to be every time more dynamic and composed by heterogeneous entities. These entities must be able to enter and exit the system, interact with each other, and adapt themselves due to environmental requirements. In the last decades, multiagent systems have contribute to model, design, and implement autonomous systems with interaction and communication capabilities. These systems are usually designed through agent societies, which facilitate the interaction, organization, and cooperation of heterogeneous agents in order to achieve different goals. In order to this paradigm be suitable for the development of the next-generation systems, features such as dynamicity and adaptability must be provided for modeling, managing, and executing agent societies.

In more detail, reorganization in agent societies provides a paradigm for designing open, dynamic, and adaptive applications. This process requires determining the consequences of applying changes not only in terms of the benefits provided, but also measuring the adaptation costs as well as the impact that these changes have on all the components of the system. The few existing approaches for reorganization mainly focus this process as responses to the society when changes occur, or as a mechanism for maximizing the utility of the system. However, it is not possible to define complex deliberation processes that obtain the best organizational configuration at each moment, based on an accurate measurement of the benefits obtained by reorganization and the costs associated to this process.

With this goal in mind, this thesis explores the area of reorganization in agent societies and focuses specifically on a novel approach for reorganization. This approach provides a decision-making support that considers reorganization in multiple organizational dimensions and is aimed at obtaining the adaptation with the highest poten-

tial for improvement in utility based on the costs of reorganization. By considering different requirements of the final configuration that is to be achieved, our approach accurately predicts the impact of the reorganization in terms of two aspects: the costs associated to carry out the reorganization process, and the benefits or costs that this process causes not only to the agents involved in the change but also to the whole system. Moreover, since several changes on different dimensions can be considered, the range of adaptation solutions is increased.

Resum

En la nova era de tecnologies de la informació, els sistemes tendixen a ser cada vegada més dinàmics, compostos per entitats heterogènies capaces d'entrar i d'eixir del sistema, interaccionar entre elles, i adaptar-se a les necessitats de l'entorn. Els sistemes multiagent han contribuït en els últims anys, a modelar, dissenyar i implementar sistemes autònoms amb capacitat d'interacció i comunicació. Estos sistemes s'han modelat principalment, a través de societats d'agents, les quals faciliten la interacció, organització i cooperació d'agents heterogenis per tal d'aconseguir diferents objectius. Amb la finalitat que estos paradigmes puguen ser utilitzats per al desenvolupament de les noves generacions de sistemes, característiques com la dinamicitat i capacitat de reorganització han de ser incorporades en el modelat, gestió i execució d'estes societats d'agents.

La reorganització en societats d'agents oferix un paradigma per dissenyar aplicacions obertes, dinàmiques i adaptatives. Este procés requerix determinar les conseqüències de canviar components del sistema, no sols en funció dels beneficis que es poden aconseguir sinó a més a més, mesurant els costos d'adaptació així com l'impacte que estos canvis tenen en tots els components del sistema. Les propostes actuals de reorganització es centren en enfocar la reorganització com una resposta de a certs canvis, o bé com un mecanisme per millorar la utilitat del sistema. No obstant això, no es poden definir mecanismes complexos de decisió que obtinguen la millor configuració dels components organitzacionals en cada moment, basant-se en una avaluació minuciosa dels beneficis que es podrien obtenir així com dels costos associats al procés.

Centrant-nos en este objectiu, esta tesi explora l'àrea de reorganització en societats d'agents i es centra principalment, en una proposta novedosa per reorganització. Esta proposta oferix un suport de presa de decisions que considera canvis en múltiples

dimensions organitzacionals, i obté l'adaptació que té associada la millora més gran d'utilitat, tenint en compte els costos associats al procés de reorganització. Considerant diversos requisits de la configuració futura que es podria obtenir, la nostra proposta prediu l'impacte de la reorganització en dos termes: els costos que es requerixen per aplicar el procés, i els beneficis i costos que la reorganització causa en tots els components de la societat. A més a més, el fet de considerar canvis en diverses dimensions organitzacionals, incrementa el ventall de possibles solucions.

Resumen

En la nueva era de tecnologías de la información, los sistemas tienden a ser cada vez más dinámicos, compuestos por entidades heterogéneas capaces de entrar y salir del sistema, interaccionar entre ellas, y adaptarse a las necesidades del entorno. Los sistemas multiagente han contribuído en los últimos años, a modelar, diseñar e implementar sistemas autónomos con capacidad de interacción y comunicación. Estos sistemas se han modelado principalmente, a través de sociedades de agentes, las cuales facilitan la interación, organización y cooperación de agentes heterogéneos para conseguir diferentes objetivos. Para que estos paradigmas puedan ser utilizados para el desarrollo de nuevas generaciones de sistemas, características como dinamicidad y capacidad de reorganización deben estar incorporadas en el modelado, gestión y ejecución de estas sociedades de agentes.

Concretamente, la reorganización en sociedades de agentes ofrece un paradigma para diseñar aplicaciones abiertas, dinámicas y adaptativas. Este proceso requiere determinar las consecuencias de cambiar el sistema, no sólo en términos de los beneficios conseguidos sinó además, midiendo los costes de adaptación así como el impacto que estos cambios tienen en todos los componentes del sistema. Las propuestas actuales de reorganización, básicamente abordan este proceso como respuestas de la sociedad cuando ocurre un cambio, o bien como un mecanismo para mejorar la utilidad del sistema. Sin embargo, no se pueden definir procesos complejos de decisión que obtengan la mejor configuración de los componentes organizacionales en cada momento, basándose en una evaluación de los beneficios que se podrían obtener así como de los costes asociados al proceso.

Teniendo en cuenta este objetivo, esta tesis explora el área de reorganización en sociedades de agentes y se centra principalmente, en una propuesta novedosa para reor-

ganización. Nuestra propuesta ofrece un soporte de toma de decisiones que considera cambios en múltiples dimensiones organizacionales, y obtiene la adaptación que ofrece la mejora más prometedora en utilidad, basándose en los costes asociados a la misma reorganización. Teniendo en cuenta diferentes requisitos de la organización que se quiere obtener, nuestra propuesta predice el impacto de la reorganización a través de dos aspectos: los costes que se requieren para aplicar el proceso, y los beneficios y costes que este proceso tiene en todos los componentes del sistema. Además, el hecho de considerar posibles cambios en varias dimensiones, el abanico de soluciones se ve incrementado.

Contents

I Introduction

1 Introduction **1**
1.1 Motivation 3
1.2 Goals 4
1.3 Structure of the document 6

II Related Work

2 Related Work **11**
2.1 Reorganization in Agent Societies 12
2.1.1 A Workshop Management System 13
2.1.2 Monitoring 14
2.1.3 Design 16
2.1.4 Selection 18
2.1.5 Evaluation 21
2.2 Approaches for Reorganization in Agent Societies 23
2.2.1 OMACS 24
2.2.2 Moise 27
2.2.3 Self-organization in task-solving environments 31
2.2.4 Autonomic Electronic Institutions 35
2.2.5 2-LAMA 38
2.2.6 MACODO 41

2.2.7 MAGIQUE 44
2.2.8 Reorganization in Agent-Organized Networks 46
2.2.9 Comparison 49
2.2.10 Other reorganization approaches 49
2.3 Discussion and Open Challenges 52
2.3.1 Monitoring 53
2.3.2 Design 55
2.3.3 Selection 58
2.3.4 Evaluation 59
2.4 Conclusions 61

III Reorganization Model

3 Organization Transition Model 65
3.1 Organization 66
3.2 Organization transition 67
3.3 Organization Transition Impact 71
3.3.1 OTI and Organization Utility 73
3.4 Multi-dimensional Transitions 74
3.4.1 Role Reallocation Transition 74
3.4.2 Acquaintance Transition 77
3.4.3 Agent Population Transition 79
3.5 Conclusions 81

4 Multi-dimensional Transition Deliberation Mechanism 83
4.1 Multi-dimensional Transition Deliberation Mechanism 84
4.2 Role Reallocation Transition 86
4.3 Acquaintance Transition 89
4.4 Agent Population Transition 90
4.5 Deliberation 91
4.6 Calculating the transition path 92
4.7 Conclusions 94

5 Reorganization Facilitator Service 95
5.1 Magentix Multiagent Platform 95
5.2 Reorganization Facilitator Service 97
5.3 Conclusions 101

IV Case Studies

6 Organization Transitions in a Tourist Application **105**
6.1 Introduction 106
6.2 Impact Measurement of Organization Transition 108
6.3 Trace 111
6.4 Performance Evaluation 113
6.4.1 Results 115
6.5 Execution 120
6.6 Conclusions 126

7 Organization Transitions on a Service Provider Network **129**
7.1 Introduction 129
7.1.1 Role Reallocation Transition Impact Estimation 133
7.1.2 Acquaintance Transition Impact Estimation 134
7.1.3 Agent Population Transition Estimation 135
7.2 Evaluation 137
7.3 Conclusions 143

V Conclusions

8 Conclusions and Future Work **147**
8.1 Conclusions 147
8.2 Future work 149
8.2.1 Dynamic Monitoring 149
8.2.2 Increasing the Range of Multi-dimensional Transitions . . . 150
8.2.3 Using Trust for Organizational Services Adaptation 151
8.2.4 Smart grid and Adaptation 152
8.3 Contributions 152
8.3.1 Indexed publications 153
8.3.2 Other conferences 154
8.3.3 Book chapters 155
8.3.4 Participation in other publications 155

Bibliography **159**

List of Figures

5.1 Magentix Architecture 97
5.2 Reorganization Facilitator Service 98
5.3 Reorganization Facilitator Interaction 99

6.1 Iterations for different agent populations 115
6.2 Iterations for different percentages of services offered 116
6.3 Time required to compute transitions 117
6.4 Events required for different agent population 117
6.5 Transition path length for different agent population 119
6.6 Interval length for transition deliberation 120
6.7 Static and dynamic organizations 122
6.8 Transitions with costs and without costs 123
6.9 Transitions by swap role and role cardinality 125

7.1 Static and dynamic organizations 138
7.2 Transitions with costs and without costs 139
7.3 One-dimensional transitions and MTDM 141

List of Tables

2.1 Reorganization phases in OMACS 27
2.2 Reorganization phases in Moise 31
2.3 Reorganization phases in TSEs 36
2.4 Reorganization phases in AEIs 38
2.5 Reorganization phases in 2-LAMA 41
2.6 Reorganization phases in MACODO 44
2.7 Reorganization phases in MAGIQUE 47
2.8 Reorganization phases in AONs 49
2.9 Comparison of reorganization approaches 54

3.1 Addition events 68
3.2 Deletion events 69

7.1 Static and dynamic organizations 139
7.2 Transitions with and without costs 140
7.3 One-dimensional transitions and MTDM 141
7.4 Transitions effectiveness 142
7.5 Algorithm Execution time 143

Part I

Introduction

CHAPTER

1

Introduction

1.1 Motivation **3**
1.2 Goals **4**
1.3 Structure of the document **6**

In the last few years, open and dynamic agent-based systems have emerged as one of the most promising areas for developing applications. In these environments, heterogeneous entities must interact, span organizational boundaries, and operate effectively within rapidly changing circumstances. Agent societies [45, 34, 32] provide a paradigm for representing the interaction of heterogeneous agents in order to reach global or individual goals, through the definition of roles, relationships, and regulations that constrain the behavior of agents inside a group of agents.

Societies have been widely used as frameworks for representing, modeling, and engineering agent-based systems, as they have also been traditionally used in human systems. It is commonly agreed that many concepts and ideas can be shared between human societies and agent-based societies [42, 92, 35]. Indeed, human societies have been used as the basis for designing and modeling agent societies [9, 89, 25].

As has been stated in the literature long ago, the organization of a system can have

a significant impact on the system's short and long-term performance [46, 92, 23, 74, 52]. However, as theorists of human societies agree, we must consider that there is not a specific society design that is suitable in all situations [60, 15]. Thus, it is important to provide societies with capabilities in order to be able to proactively or reactively anticipate or react to different variables [53] by means of reorganization.

Similar to the importance that reorganization have in human systems, reorganization has a remarkable importance in agent societies. According to [15], agent societies can be adapted by changing the roles, knowledge, and activities of agents to suit new problem situations. This process requires to identify dysfunctional organizational issues, to provide solutions which solves these issues, and to have mechanisms for managing transitions among organizations.

Reorganization can be viewed as the mechanism that modifies the structure and behavior of the agent society, such as adding, removing, or substituting components, which are done while the system is running and without bringing it down [36]. In this sense, reorganization in agent societies defines a process that changes an organization into a new one [55]. These changes are related to the organization specification such as roles, goals, services, norms, and the agent population as well as changes in the relationships among these components.

New applications are aimed to be composed by collaborative and heterogeneous entities to be dynamic and self-organizing [99, 79]. Specifically, due to the increase in the number of open agent-based applications, adaptive societies that adjust themselves to gain advantage in their current environments are likely to become increasingly more important over the next few years [57, 70]. Reorganization eliminates the need to determine all possible runtime conditions a priori, which is unknown in many systems. Before this can occur, the space of organizational options must be mapped and their relative benefits and costs understood [52].

1.1 Motivation

In the last few years, some approaches have been developed for representing and modeling reorganization in agent societies. This support is usually provided by means of frameworks and infrastructures which manage the reorganization process. By providing agent societies with capabilities for discovering, evaluating, and representing issues related to reorganization, a process of organizational self-design can be facilitated, in which a system automates the process of selecting and adapting an organization dynamically [28, 52, 88]. However, as we will see in the next chapters, current approaches present some limitations that should be overcame for the next generation of open and dynamic systems.

Most of the current reorganization approaches usually focus on changes in a specific dimension of the agent society such as the roles played by agents, the relationships between agents, or the norms of the system. However, reorganization in agent societies should involve changes that are focused on different dimensions such as organizing structures, coordination mechanisms, or work practices [82, 6]. A reorganization approach that considers several dimensions to be adapted at the same time would increase the range of adaptation possibilities. This approach would require a greater level of integration between all the changes that can occur and their consequences. This multi-dimensional reorganization must require to evaluate the interdependences of changes that could be applied simultaneously.

In addition, a reorganization process should provide some kind of increase in utility [36]. However, this utility should take into account not only the gain in utility but also the costs of carrying out the reorganization. Related to this issue, current approaches for reorganization do not provide mechanisms for accurately measuring the impact of a reorganization process. As an example, some of them assume that a relationship modification between a pair of agents or a change in the role played by an agent do not have implications in the rest of the population and can be carried out without requiring additional costs (time, resources, etc.). However, as in human societies, not

every change in an agent society has the same implications in terms of costs or has the same impact in the whole society. Therefore, complex deliberation processes that evaluate the suitability of the reorganization before carrying out the process must be provided. This evaluation must include not only the benefits produced by the reorganization but also how these changes would affect other agents of the organization and the costs for carrying out the process.

Therefore, we can summarize the motivation of this thesis as the needed of a reorganization approach that considers reorganization in multiple organizational dimensions and is aimed at obtaining the adaptation with the highest potential for improvement in utility based on the costs of reorganization. This approach must provide support for an organization to change dynamically according to the requirements of the system, by considering different organizational elements to be changed. In addition, the underlying mechanism to deliberate the best reorganization to be carried out, must provide an accurate measurement of the reorganization implications (positive and negative). These evaluations must consider not only the agents involved in the change but also how a change can influence the performance of the rest of the agents of the society, as well as the costs associated to carrying out the reorganization process itself.

1.2 Goals

In order to fulfill the requirements defined in previous section, we must provide a reorganization approach that overcomes some of the limitations of current approaches which constraint the development of next-generation of open and dynamic systems. Specifically, this reorganization approach is aimed at improving the quality of the reorganization process and the accuracy of the reorganization consequences. In order to carry out this goal, first we will establish the requirements to be provided by the reorganization approach by analysing the current state of the art of the most relevant approaches related to reorganization in order to determine the advantages and limi-

tations of current proposals. After developing this reorganization approach, we will validate the performance and the effectiveness of this approach by using two case-studies. Following we present in more detail the goals and sub-goals of this thesis:

1. Analysis of current approaches that support the specification and execution of reorganization in agent societies. We contribute an analysis of the current state of the art regarding reorganization in agent societies. This analysis is aimed at providing a detailed landscape of the most relevant approaches, in order to define the main advantages and limitations of each approach. This will help us to determine which requirements should be provided in the next generation of reorganization approaches in order to overcome the limitations of current approaches.

2. Design of the reorganization approach aimed at providing a multi-dimensional support for reorganization in agent societies, which obtains an accurate measurement of the reorganization consequences. This reorganization approach will consist on (1) providing a reorganization model which determines the elements needed for reorganization in different dimensions; (2) providing a design of the mechanisms required for reorganization and deliberation; and (3), providing the implementation of these mechanisms by means of an infrastructure integrated in a multiagent execution framework. This goal can be divided in the following sub-goals:

   (a) Design of the reorganization model which supports the dynamicity required for reorganization according to different changes that are considered in this process. This model must define how an organization can evolve to a different organization and the specification of the costs related to the reorganization focused on several dimensions. This would require to express how an organization can be changed into another one depending on the specific changes that are carried out.

   (b) Design of a reorganization mechanism for deliberation, which provide the support to calculate which is the most beneficial organization that

can be achieved according to specific reorganization requirements. This mechanism must provide the algorithms that compute the costs and the benefits for this organization and the sequence of steps required to carry out this process.

(c) Implementation of the reorganization mechanism for deliberation and reasoning in a multiagent framework. The algorithms designed must be implemented in a multiagent framework in order to organizations to be able for using this reorganization mechanism.

3. Evaluation of the model, the proposed mechanisms, and the implemented infrastructure. This goal can be divided in the following sub-goals:

   (a) Empirical evaluation to validate the model and test the efficiency, scalability, and performance of the proposed algorithms. This evaluation is carried out by using synthetic data.

   (b) Evaluation of the reorganization mechanism in order to prove the effectiveness of the multi-dimensional approach. In order to carry out this evaluation, we focus experiments on testing the three basis of the approach: evaluation of the adaptability support, evaluation of the cost-aware support, and evaluation of the multi-dimensional support.

1.3 Structure of the document

In order to reach the goals and sub-goals defined in previous section 1.2, the rest of this document is structured as follows:

- In Chapter 2 we present an analysis of the most relevant works related to reorganization in agent societies, which corresponds to the goal 1.
- In Chapter 3 we present the reorganization model, called the Organization Transition Model, which corresponds to the sub-goal 2(a).

- In Chapter 4 we present the reorganization mechanism, called the Multi-dimensional Transition Deliberation Mechanism, which corresponds to the sub-goal 2(b).
- In Chapter 5 we present the infrastructure that supports the mechanism, which is implemented as the Reorganization Facilitator Service, which corresponds to the sub-goal 2(c).
- In Chapter 6 we present a real-world example to evaluate the reorganization model, which corresponds to the sub-goal 3(a) and 3(b).
- In Chapter 7 we present a real-world example to evaluate the reorganization mechanism, which corresponds to the sub-goal 3(b).
- In Chapter 8 we present the conclusions of this work, the contributions in terms of publications, and some future research lines related to this work.

Part II

Related Work

CHAPTER

2

Related Work

2.1 Reorganization in Agent Societies 12
2.2 Approaches for Reorganization in Agent Societies 23
2.3 Discussion and Open Challenges 52
2.4 Conclusions 61

Being the reorganization in agent societies an important research focus in the last few years, it is relevant to analyze how current reorganization approaches provide support to agent designers in order to develop adaptive agent societies. The goal of this Chapter is to describe in detail some of the most relevant existing approaches, in order to show the advantages and limitations of each one. In addition, related specially to these limitations, we can define a long-term goal related to define which research issues would be interesting to be addressed in the next future years, in order to develop agent societies that autonomously adapt and regulate themselves in response to events and changes in the environment. First, we describe what is reorganization in agent societies and different parameters that are useful to compare different approaches along the reorganization life-cycle. Second, we detail some of the most relevant works related to reorganization according to the parameters remarked above. Finally, we outline some open challenges that can be found due to this analysis.

2.1 Reorganization in Agent Societies

The concept of agent society can be slightly different depending on the authors. However, common properties can be found along the literature [45, 27, 37, 8]. According to these, we can view an agent society as a flexible and robust group of interacting agents having common goals. Within the society, agents have different roles which require a set of competencies. These agents can interact with each other, identify their abilities, and request activities on behalf of others, being observed to respect social norms. Agent societies provide the basis for designing complex, structured, organized, and regulated systems. Depending on the application and the problem, each agent society model defines different requirements and uses different names to represent some of the above elements.

Reorganization in agent societies can be defined as a process that changes the society [55]. This reorganization is referred to modifications in the structure and behavior of the agent society, such as adding, removing, or substituting components, which are done while the system is running and without bringing it down [36]. These modifications are related to the organization specification, i.e., roles, goals, services, norms, and the agent population, as well as changes in the relationships among these components. In this respect, the life-cycle of a reorganization can be defined as the process of analyzing the problems of the current agent society, proposing adaptation solutions, selecting and implementing a reorganization, and evaluating this process once it is applied.

Authors agree that reorganization [101, 62] and more specifically, reorganization in agent societies [91, 55], can be represented as a loop process composed by different phases. The specific definition of each phase may slightly change from one author to another. We try to use a general definition that can be adjusted to different models of agent societies, in order to cover a wide range of applications. The reorganization process starts by allowing the system to perceive information from the managed elements (resources, agents, etc.). This information is analyzed in order to assess

that changes are required and one or more set of changes (adaptation solutions) are proposed. Then, a specific solution is selected and needed to be implemented, which consists of applying the changes that are required. Finally, the reorganization process is evaluated and the loop is closed by starting again the starting phase. According to this, we define the following main phases for representing the life-cycle of a reorganization process: monitoring, design, selection, and evaluation. Frameworks that support reorganization in agent societies implicitly define issues that are related to these phases. In order to discuss the features of the most relevant approaches, we use these phases to analyze the support provided by these approaches for these phases and how this support is implemented. In the following section we introduce a running example in order to support the discussion of these phases. Then, we present a detailed description of each phase.

2.1.1 A Workshop Management System

As a simple example to illustrate an agent society, let us consider a system for supporting the process of producing the technical program for an international workshop, which is similar to other examples used for illustrating adaptive systems [73, 25]. In this example, agents are associated to the actors involved in the process and play some specific roles inside this society (PC chairs, PC members, reviewers, etc.). Depending on the role or roles played by each agent, some services or capabilities must be provided. As an example, an agent playing the PC chair role must provide some management skills, while an agent playing the reviewer role must provide some knowledge to evaluate the submitted papers. The objective of the agent society can be defined as organizing the workshop, while agents or roles can be assigned to fulfill some sub-objectives such as the paper evaluation or the paper selection.

Interaction between agents inside the society can follow some structure, according to the interaction patterns that are defined for communication. As an example, reviewers can only interact with the PC chairs, not allowing direct interaction between two reviewers. In addition, some regulations or norms can be defined at different levels

in this system. An example of these regulations can represent that a reviewer cannot evaluate a paper that is not assigned to himself.

The objective of this example is to help the reader to better understand all the features that are defined along the paper, by providing a realistic real-world example. Depending on the different approaches some of the requirements of this example would be represented in different ways.

2.1.2 Monitoring

The monitoring phase defines the problems of *why* and *when* a society needs to be adapted. Monitoring is essential in order to be able to detect undesirable behavior that needs to be corrected [48], which can be triggered by changes in the environment. Theorists of human societies [13, 98, 53] define two possible categories of strategy options for reorganization: environmental determinism and organizational choice. The first option corresponds to reactive changes that are triggered by the social interaction with the environment. The second option refers to the social ability to create a voluntary response in terms of adaptation. According to these strategy options, we propose to classify the monitoring phase in agent societies by following a terminology that is more widely used in agent societies [36].

A *reactive* strategy occurs when the agent society automatically responds to events that cause a reorganization such as the addition or deletion of a new role, agent, etc. These events cause the agent society to make the required adjustments in order to continue to fulfill its goals. As an example, in the workshop management system, a reactive strategy could detect that some paper needs to be reallocated due to its reviewer is not available anymore. Approaches that follow this kind of reorganization focus on the need for an adaptation process to be guided by events. In contrast, a *proactive* strategy requires an implicit mechanism for reasoning about the current situation in order to decide that a reorganization is required. In the example of the workshop management systems, a proactive strategy could decide that a paper needs

to be reallocated in order to assign it to a reviewer which is more related to the topic.

Related to these strategies, the reorganization logic is used to define that an adaptation is required. This reorganization logic can be *predefined*, if the events that trigger this adaptation or the mechanisms for deciding that a reorganization is required are implemented at design time. Otherwise, this logic can be *adaptable* if it can be changed when the system is running. In the workshop management system, this logic could represent that the deadline must be extended if the number of submitted papers do not reach a minimum threshold. In a predefined logic this threshold could not be changed at runtime, while in an adaptable logic this could be changed depending on the number of papers already submitted.

According to other works related to reorganization in agent societies [48], the information that is acquired can be specified and used *off-line* or *on-line* in order to improve the system's behavior. An off-line specification defines the information to be monitored at design time and cannot be changed; in contrast, an on-line specification, the information that is monitored can change depending on the requirements of the system during execution. Similarly, the information that is monitored can be used off-line or on-line.

Finally, in human societies, the decision-making process can be carried out by individuals or by their institution [53]. Similarly, we define two different implementation types for the monitoring phase: a *centralized* way if an agent or a specific authority is responsible for deciding that a reorganization is required; a *distributed* way if a pool of agents can decide that a reorganization is required either autonomously or by means of an agreement. A support for monitoring should be provided by any approach that supports reorganization. The greater dynamicity provided by this support to implement the monitoring mechanisms, the more flexibility would be provided to develop adaptive agent societies.

2.1.3 Design

The design phase defines the problem of *how* a reorganization is carried out. Once a reorganization process is required, the design phase includes an analysis of the organizational elements and a reorganization proposal that changes specific elements of the agent society. Similar to the monitoring phase, design can also be carried out in a *centralized* way if a single agent or a central authority is responsible for proposing the reorganization solution. A *distributed* design involves the participation of several agents in the reorganization solution proposal.

Depending on the specific model, current reorganization approaches provide support for changing different elements of the agent society based on the requirements of the problems that they consider.

Some authors propose a classification of reorganization types in behavioral and structural adaptation [36]. Nevertheless, a more detailed classification can be provided by including common types of changes that can be found in the literature according to the following dimensions:

- **Open System** support allows changes in the agent population, i.e., agents can enter or leave the system. As an example, in the workshop management system this would be referred as allowing the entrance of new reviewers into the system due to the number of submissions are higher than expected.
- **Emergence** support allows elements that define the social behavior to be changed; i.e., the addition or deletion of the roles that agents can play, social goals, etc. As an example, in the workshop management system this would be referred as allowing the creation of a new publicity chair role, which is required to disseminate information about the workshop to the largest possible appropriate technical audience.
- **Behavioral adaptation** support allows changes related to the behavior of the agents that populate the society. For example, this involves changes in the ca-

pabilities offered by an agent in order to be able to play a role. As an example, in the workshop management system this would be referred as allowing agents to add new topics in which they are experts.

- **Functional adaptation** support allows changes in how different elements of the agent society are related to each other, which affects the society functionality, such as changes in the services offered by a role or changes in the roles that agents play. As an example, in the workshop management system this would be referred as allowing the reassignment of a paper to another reviewer when the previously assigned reviewer is not able to provide its reviews by the deadline.

- **Structural adaptation** support allows changes in the relationships between elements of the agent society, which affects the social structure, such as relationships among the agents. As an example, in the workshop management system this would be referred as allowing the interaction between two reviewers which were not previously allowed to interact to each other, in order to discuss the acceptance of a given paper.

- **Norm adaptation** support allows changes in the regulations of the agent society. This support can be related to modifications in the specification of the norms that govern the agent society as well as the addition or deletion of new norms. As an example, in the workshop management example this would be referred as extending the notification deadline due to several requests by reviewer agents.

Depending on the elements that are allowed to be changed in a reorganization process, a wide range of different solutions can be provided. Therefore, it would be desirable for a reorganization to be able to consider as many dimensions as possible. As we will see in the following sections, some reorganization approaches focus on changes in an specific dimension, while other approaches are more flexible by considering changes in several dimensions.

2.1.4 Selection

The selection phase defines the problem of choosing *which* reorganization is finally implemented. If several reorganizations have been proposed in the design phase, the selection phase determines which of these proposals is applied. Similarly to previous phases, if a single agent or a central authority is responsible for this selection, we consider the selection phase to be *centralized*. In contrast, if several agents are involved in the selection phase (for example, by a negotiation process or by social choice), we consider the selection phase to be *distributed*. Depending on whether a single reorganization is designed or several designs are proposed, several criteria can be used in the design phase to guide the design, or in the selection phase to guide the selection.

In some scenarios, reorganization can be viewed as the mechanism that allows the society to achieve the social goals. As an example, this occurs when some event prevents the society to achieve its goals, and therefore, a reorganization is required to achieve a goal fulfillment. In other scenarios, according to Dignum et al. [36], reorganization is desirable if it leads to increase the utility of the system. These authors define two kinds of utilities: individual and social. Individual utility is different for each agent, while social utility can take into account the individual utility of each agent. Nevertheless, in some scenarios, an individual utility increase may not cause a direct social utility increase. What is more, an individual utility increase may have a negative impact in the society. Therefore, it is relevant to analyze whether the utility that is taken into account for the reorganization is referred to the direct benefits caused in the agents involved in the change, or whether the utility also considers indirect benefits, which are referred to other agents of the society.

The utility must take in account both the reorganization success and the cost of any change needed to achieve the reorganization from the current situation [45, 5]. Actually, other works such as Cheng et al. [26] state that the reorganization process must be also evaluated, e.g. in terms of its impact in space of time.

As stated in [67], most organizational changes may encounter problems: they often take longer than expected and desired; the cost of managerial time may be increased; and there may be resistance from the people involved in the change. In order to deal with this problem, we consider that both the benefits obtained by reorganization and the costs associated to this process are important aspects that should be taken into account in order to define the suitability of a reorganization process.

According to the above criteria, we classify the dimensions used for the selection phase depending on the issues that are considered for reorganization. We include benefits and costs in separated dimensions in order to provide a global view that can be applied to different approaches. Both benefits and costs can be referred to time, money, resources, and so on, depending on which is the society focus:

- The **Goal fulfillment** takes into account the fulfillment of the social goals in order to select the reorganization that is implemented. As we stated above, reorganization in this case is only focused on achieving the society's stability by fulfilling the social goals. As an example, in the workshop management system, a selection focused on the goal fulfillment could be triggered when the agent associated to the PC chair role is not available to achieve its goals and the system needs to reallocate another agent to this role.
- The benefits of the agent society in order to design or to select a reorganization can be classified as:
  - **Direct benefits** represent the benefits that are associated to the individual elements involved in the change. As an example, in the workshop management system this could be referred as how the reviewer a_x is positively affected if it is reallocated to review a different paper, e.g. if the new assigned paper if more related to its research topic, its revision would be more profitable to the author.
  - **Indirect benefits** represent the benefits that are associated to other elements that are not involved in the change but that can be influenced by

this change. As an example, in the workshop management system this could be referred as how the reviewer a_y is positively affected if the reviewer a_x is reallocated to a different paper, e.g. if the reallocation of a_x causes that the paper assigned to a_y is changed by the paper previously assigned to a_x, which is more related to its research topic.

- Similarly to the benefits, costs can be divided in:
  - **Direct costs** represent the individual costs that are associated to the elements involved in the change. As an example, in the workshop management system this could be referred as how an agent a_x is negatively affected if it is reallocated to review a different paper, e.g. if the new assigned paper if less related to its research topic, its revision would be less profitable to the author.
  - **Indirect costs** represent the costs that are associated to other elements that are not involved in the change but that can be influenced by this change. As an example, in the workshop management system this could be referred as how a different agent a_y is negatively affected if a_x is reallocated to a different paper, e.g. if the reallocation of a_x causes that the paper assigned to a_y is changed by the paper previously assigned to a_x, which is less related to its research topic.

In addition, as we stated above, other costs related to the reorganization process can be considered:

- **Reorganization costs**, which allow a representation of the costs required to carry out the reorganization process, i.e. to apply each change associated to the reorganization. As an example, in the workshop management system this could be referred as the time required to reallocate a new paper to the reviewer a_x.
- **Computation costs**, which allow a representation of the costs required to compute or select the reorganization. These costs are usually represented

as the time required to design a reorganization and select a solution. As an example, in the workshop management system this could be referred as the computation cost required to calculate that a_x is reallocated to review a new paper.

Depending on how some of these criteria are considered for reorganization, the consequences of reorganization can be accurately estimated to a greater or a lesser degree. These consequences refer to how a reorganization influences the fulfillment of the goals of the system, how the reorganization influences the agent society utility (i.e. how beneficial it is for any agent of the society), and how costly the reorganization is (i.e. how each agent could be affected and how costly the process to be carried out is). Therefore, the more criteria is considered for selecting the reorganization the more specific the reorganization impact can be estimated.

2.1.5 Evaluation

Reorganization approaches should also encompass techniques for monitoring and controlling the system once the reorganization is deployed [70]. The evaluation phase defines the problem of analyzing *how well* a reorganization has been performed. This phase provides feedback from the reorganization in order to assess whether or not the reorganization was as expected. This allows to evaluate the quality of the reorganization that was designed and selected as well as the quality of the society that was achieved, in order to take it into account for future reorganizations, which increases the quality of future reorganizations.

Researchers from human societies stated the relevance of a feedback process in order to achieve a successful reorganization [12, 11, 19]. This feedback helps to improve the control of new environments. According to these authors, several properties are related to this process such as the frequency of the feedback, the method (by inquiry or in time-periods), the individual that provides the evaluation, and the information

or topic that is provided. Other works related to adaptive systems such as [26] define different factors that measure the criticability of the reorganization, the predictability, the overheads associated to it, and whether the system is resilient in the face of change.

All of these properties define evaluation processes. However, we propose a general classification in three general dimensions in order to differentiate which is the aim of the information obtained by this evaluation. These dimensions allow to classify a wide range of approaches depending on which of the previous phases (monitoring, design, and selection) the evaluation provides the feedback for. The techniques in which the evaluation is supported by each reorganization approach will be discussed and analyzed in the following sections.

- An evaluation of the reorganization **proposal** is carried out when the approach analyzes the reorganization that has been designed depending on the requirements of the problem in order to consider this information for future reorganizations. This dimension provides an evaluation that is related to the design that has been carried out, allowing past decisions to be considered in the future, if the reorganization requirement is similar. As an example, in the workshop management system this could be referred as evaluating which agents have been reallocated to which papers in order to consider this reallocation in the future, if the reorganization requirements are similar.
- An evaluation of the reorganization **process** is carried out when the approach analyzes the reorganization execution in order to improve the predicted implementation of the process. This refers to issues such as whether or not the time and resources used were as expected during the implementation, if a setback appeared during this process, etc. This dimension provides an evaluation that is related to the selection that has been chosen, allowing the approach to improve the accuracy for estimating the consequences of future reorganizations, and therefore, improving the selection process. As an example, in the workshop management system this could be referred as evaluating if the reorganization

costs for reallocating agent a_x to review a new paper were as expected or in contrast, if this agent was reluctant to change.

- An evaluation of the **future state** of the society is carried out when the approach analyzes the future state of the society that is achieved, in order to improve the predicted performance of the future society. This dimension provides an evaluation that is related to the monitoring process, allowing to improve the accuracy for estimating the performance of future modifications. As an example, in the workshop management system this could be referred as evaluating if the reallocation of agent a_x to review a new paper caused the expected benefits and costs (e.g. if the deliberation process delay was as fast as expected when the reallocation was computed).

Similarly to previous phases, the evaluation phase can be *centralized* if a single agent or a central authority is responsible for this phase, or *distributed*, if several agents are involved in this process. The greater amount of information related to the level of success of the reorganization is incorporated into the system, the more learning capabilities can be associated to the process for gaining experience from past reorganizations.

2.2 Approaches for Reorganization in Agent Societies

In this section, we provide an analysis of some relevant approaches to discuss their suitability in dealing with reorganization in agent societies. For each approach we describe their support for each phase of the reorganization life-cycle as well as some critical considerations. In Section 2.2.9 we show a graphical comparison between all of them in Table 2.9.

According to these criteria, several approaches proposed in the literature are only focused on specific phases or changes in specific dimensions. As an example, approaches such as [81, 90, 50] focus on the selection of the best role reallocation,

while other works such as [59, 97] are focused on the selection of the best structural reorganization. Other works are especially focused on the monitoring phase [48] or in learning algorithms to optimize the agent interactions [3]. In addition, other approaches require a high human interaction in order to choose the reorganization decision [24].

Other well-known approaches can be analyzed since the perspective of several phases and dimensions. In the following, we analyze some of these recognized approaches.

2.2.1 OMACS

OMACS (Organization Model for Adaptive Computational Systems) [31] (Table 2.1) is a metamodel for defining the reorganization at runtime in order for an agent organization to be able to achieve its goals effectively. The society model used in this approach is an organization composed by goals, roles, and agents along with additional entities called capabilities, assignments, and policies. Each role is defined to achieve a particular goal or a set of goals. Capabilities determine which agents are assigned to which roles according to the set of capabilities that are required to play each role and the capabilitites possesses by each agent. Assignments define a set of agent-role-goal tuples $\langle a, r, g \rangle$ to indicate that an agent a is assigned to play the role r in order to achieve the goal g. Finally, policies specify the regulations of the system such as "one agent may only play one role at a time".

2.2.1.1 Monitoring

Reorganization is carried out reactively way when an event occurs and changes the state of the organization, which triggers the reorganization [75, 76, 73]. The agent that is responsible of realize these changes is the organization master, which is a specialized agent that possesses complete information about the organization and which is able to execute reorganization algorithms. The information that is required by the organization master is specified at design time but it is used on-line.

A specific kind of reorganization policy is used to describe the reorganization logic at design time. This logic defines rules that represent direct actions that are taken in order to trigger the reorganization. An example of reorganization policy can express that "if agent a_x is playing role r to achieve the goal g, and a_x becomes incapable of playing it, then if agent a_y is capable of playing r, it should be assigned to goal g and a_x should be de-assigned". This reactive approach can increase the reasoning efficiency in anticipated scenarios.

2.2.1.2 Design

Reorganization design is carried out as a centralized process that changes the assignment of agents to roles when different events occur. Current implementation considers two types of events that change the state of the organization. On the one hand, changes in the goal set cause a reorganization: an insertion of a new goal, a goal achievement, and a goal failure. On the other hand, changes in agents also cause a reorganization: if an agent is removed from the organization, and if an agent loses a capability that negates its ability to play a role that it is assigned.

These events are specified before running the system and are responsible for causing reorganization, but they cannot be used in the design phase of the reorganization process. Therefore, the changes considered in the design phase define functional adaptation. Once a trigger occurs, general-purpose reorganization algorithms are implemented to find the appropriate assignments, which determine the design of the solution [102].

2.2.1.3 Selection

In order to obtain the best set of assignments that maximizes the organization's ability to achieve its goals, functions are defined to evaluate how effective a role is in achieving a specific goal, and how effective an agent is in providing a capability. These functions return a real value that determines this effectiveness. Thus, an assignment

determines how well an agent can play a role to achieve a goal. The organization assignment function computes a score that represents the *goodness* of the organization [30]. This is usually calculated as the sum of the scores of all the assignments. We must also point out that changes in the effectiveness of agents are not considered.

In the algorithms that are used for determining the appropriate assignments, every combination of goals, roles, and agents must be computed. In order to avoid this, the authors propose using assignment policies that restrict the range of valid solutions, such as restricting the number of roles played by a single agent. In this approach, if an optimal reorganization is found, this is automatically selected and implemented by the organization master in a centralized implementation, who sends the new assignments to agents . As an example, in [74] authors present a reorganization process that is approached as a role reallocation that determines which agent plays which role.

2.2.1.4 Evaluation

The OMACS approach does not provide support to measure the execution of the reorganization in terms of how the process can finally be carried out or whether or not the reorganization has achieved the expectations. Since the monitoring phase is carried out by policies, metrics could be used to allow designers to make design-time tradeoffs between flexibility and computational costs. Nevertheless, there is no specific definition of costs associated to the reorganization process itself. Therefore, these metrics are difficult to measure at design time without any knowledge of how the organization will behave at runtime.

OMACS specifies different scores associated to relationships in order to provide a measurement for organization utility. The optimal assignment refers to the organization assignment function that has the highest score. However, the costs of applying changes and the impact that these changes have on the rest of the agents cannot be specified in this approach. It is assumed that an agent a being reallocated to play a role r can be carried out with a non-associated cost. Moreover, this assignment

Table 2.1: Reorganization phases in OMACS

	OMACS
Monitoring	Reactive reorganization triggered by events. Centralized implementation by the organization master. The information that is monitored is specified at design time but is used on-line. The reorganization logic is predefined and cannot be changed.
Design	Centralized algorithms that determine the best set of assignments. Functional adaptation: changes in the assignment of agents to roles.
Selection	If a new best set of assignments is obtained, this is automatically selected by the organization master and implemented by sending this information to the agents involved in this process. The best set of assignments fulfills the organization goals and maximizes the organization utility.
Evaluation	There is not any evaluation implemented.

would not have any effect (positive or negative) on the rest of the population. By considering reorganization costs (material costs, resources, time, etc.), the costs for achieving the optimal assignment could be high and may not be worth the benefits obtained. In contrast, a sub-optimal assignment might be achieved with a lower cost, increasing the value of the overall process.

2.2.2 Moise

The Moise reorganization approach proposed by [55] (Table 2.2) is aimed at providing support in order to adapt an agent organization to its environment and to help it to efficiently achieve its goals. The society model used in this approach is Moise+ [56]. This model defines an organization which is composed by agents, roles, missions, and the deontic dimension. Each role represents a set of constraints that an agent follows when it plays this role. These constraints represent the structure dimension (relations between roles) and the functional dimension (missions, deontic dimension). A mission is a set of coherent goals that an agent can commit to. The deontic dimension specifies the permissions and obligations of a role in a mission.

In order to carry out the reorganization process, specific roles are defined. The organization manager role is played by an agent, which is in charge of managing the

reorganization process. This agent has complete information about the current state of the organization and has permission to change it. The historian role is defined to maintain information regarding the entire history of the organization. This information could be useful for the monitoring and design phases. An agent that plays this role informs the organization manager of all the social events that it has participated in. The monitor role is in charge of identifying situations that require a reorganization. Finally, the adaptation expert role is played by agents that are in charge of identifying current problems of the organization and proposing reorganization solutions.

2.2.2.1 Monitoring

In the Moise reorganization approach, reorganization is a proactive process that changes the current state of the organization into a new one [55]. The monitoring phase is implemented in a distributed way by monitor agents. These agents are able to decide that a reorganization is required based on their internal knowledge. The information required to be monitored is specified off-line but is used on-line without stopping the execution. The logic for reorganization is implemented at design time and cannot be changed during runtime. As an example, a monitor agent can realize that a reorganization is required since a request cannot be satisfied due to the production rate going below a threshold, etc.

2.2.2.2 Design

A wide range of possible changes is defined: the roles played by agents, the number of agents playing a role, the parameter related to an obligation, etc. When a reorganization is required, the organization manager is able to invite adaptation expert agents to propose design solutions. These agents are in charge of providing a plan of changes that modifies the current organization into a new organization. The plan of changes is composed by individual events such as adding a specific role, a mission,

etc. Thus, the design phase can also be implemented in a distributed way.

2.2.2.3 Selection

In the case that several designs are proposed for reorganization, the organization manager is mainly responsible for carrying out the selection of the changes to finally be implemented according to the own methods of this manager. The main problem is to define the criteria for selecting the most promising proposal, which could be carried out individually by the organization manager or also in a distributed way with the participation of other agents. As an example, in [54], a voting system between experts is used to determine the design that is finally going to be implemented. In [55], a Q-Learning algorithm is used to find out the decision policy, which is used in the selection phase. The selection implemented in the examples provided considers designs with the most promising improve in the benefits caused in the agents involved in the change. Some costs are also considered such as the number of roles that are required to be changed.

2.2.2.4 Evaluation

An evaluation phase is not explicitelly defined in the Moise reorganization approach. However, since the historian agent maintains information of the organization lifespan, specific information about the successful performance of the organization that is achieved after reorganization, can be used by the organization manager for future reorganizations [54]. There is no support for measuring how the organization has been carried out in order to estimate the success of future reorganization processes. Again, the use of agents to support these phases makes the approach so general that specific solutions must be implemented by the agent designer.

Note that, this reorganization approach provides great flexibility and can be used in a wide range of applications since individual agents are in charge of carrying out the reorganization phases. Different methods can be implemented at the agent level

depending on the domain. Constraints regarding the monitoring, design, and selection phases are defined according the application requirements. Since several agents may participate in the monitoring, design, and selection phases, different mechanisms can be used to provide heterogeneous design solutions and also to select these solutions, such as case-based reasoning, learning, negotiation, etc [2, 29]. However, this flexibility can become a drawback since specific methods are not provided by the reorganization approach itself to carry out these phases. Specifically, methods for designing solutions are not provided since they are assumed to be under the control of the adaptation expert agents. If this behavior is implemented at design time, the information regarding how the organization is performing at runtime cannot be included. Moreover, methods for measuring the goodness of a reorganization are not provided by the Moise reorganization approach. This goodness should be represent the impact that the reorganization would cause in the organization (direct and indirect benefits and costs) and the costs for applying the reorganization. This support must be implemented at the application level. As an example, in [54], three criteria are chosen for selecting a solution: the experience of each expert in past reorganizations, the success of the proposals of each expert in past reorganizations (which are obtained by the Historian agent), and the cost of the proposal in terms of global costs related to how many missions and roles would be deleted if the proposal is implemented.

The concept of plan of changes provided by reorganization expert agents has two main advantages. The first advantage is that defines step by step how the organization specification should be changed. Thus, when an expert proposes a plan of changes, implementation issues also have to be dealt with (add the role r_x and afterwards remove the role r_y, remove the role r_y and afterwards add the role r_x). The second advantage is the possibility of changing only some part of the organization.

As a general conclusion, we consider that the Moise reorganization approach has the advantage that since many agents are used to provide the different phases of the reorganization process, many different reorganization solutions can be used, and, therefore, a better reorganization decision can be taken. However, the main disadvantage

Table 2.2: Reorganization phases in Moise

	Moise
Monitoring	Proactive reorganization carried out in a distributed way by monitor agents. The information that is monitored is specified off-line and it is used on-line. The reorganization logic is predefined and cannot be changed.
Design	Distributed design is carried out by expert agents. A wide range of changes is allowed: open system (new agents can enter in the organization), emergence (new roles can be created), functional adaptation (the assignment of roles to agents), structural adaptation (the relationships between the agents), normative adaptation(the parameter related to an obligation).
Selection	The organization manager carries out individually the selection of the proposal or by requesting other agents. The direct benefits caused in the organization are usually considered in the selection phase. Some global costs such as the number of missions that are deleted are used as well.
Evaluation	The historian agent can be used for retrieving information regarding how was the performance of the future organizations after reorganization. In some example provided, learning techniques are used to evaluate how long a solution has been valid and which has been its performance, in order to take into account past decisions for future reorganizations.

of this approach is that all the reorganization phases must be implemented at the agent level. Thus, the agent designer must provide his own methods and tools for the specific application.

2.2.3 Self-organization in task-solving environments

Kota et al. propose a self-organization approach that is mainly focused on task-solving environments (TSEs) [64, 65, 66] (Table 2.3). Specifically, this approach has the following properties: the reorganization process is continuous, is carried out internally, and has no central control. The society model used in this approach consists of an organization of cooperative agents that are in a TSE. Agents receive tasks, execute actions, and return a result. A TSE presents a dynamic stream of tasks that have to be performed. These tasks require services to be processed, which are provided by agents. Agents need to interact with one another in order to access services provided by other agents. In order to do this, the agents are connected with each other accord-

ing to different levels of relationships: acquaintance, peer, and superior-subordinate. These relationships define the structural topology of the organization.

In this approach, reorganization consists of a process that changes the structural topology of the society in order to increase the performance. These type of structural changes is a common adaptation in other kind of general networks [93, 51]. The TSE approach is aimed at enabling each pair of agents to continuously and autonomously evaluate (and change if required) their relationships based on past interactions.

2.2.3.1 Monitoring

Reorganization phases are carried out in a distributed way by each pair of agents. The monitoring phase is proactively carried out by each pair of agents, which evaluate their relationship by taking into account their history of interactions. The internal logic for reorganization is provided at design time. The information that is monitored is used on-line without stopping the execution and is also specified on-line since it depends on the relationships of each agent at a specific moment.

Apart from this proactive reorganization, a reactive reorganization is also supported. Agents can enter or exit the organization and can change their capabilities at predefined times. When these events occur, a reorganization is reactively triggered. Similarly to the proactive reorganization, the changes allowed to overcome these situations are related to the agent relationships. When an agent enters the organization, it needs to be related to other agent/s. In this case, the reorganization is carried out by evaluating the past interactions of the agent with other agents and predicting the utility with other agents that they have not previously have relationships with. A reorganization is also carried out reactively when an agent leaves the organization. With regard to behavioural adaptation, the gain or loss of services by agents is another kind of reorganization that is also considered. When this occurs, a reorganization is carried out by measuring the historical interactions of the agents involved in the process.

2.2.3.2 Design

Every pair of agents chooses actions to establish or dissolve their relationship in order to improve the utility function associated to this relationship. Based on their current relationship, different modifications (form peer, remove peer, form subordinate, remove subordinate) can be carried out. These represent different reorganization alternatives proposed in the design phase.

2.2.3.3 Selection

Each alternative of relationship modification has an associate utility function that is calculated as the expected benefits and costs associated to this alternative. Each pair of agents selects and implements the modification that maximizes the utility function of their relationship. Specific mechanisms are provided for evaluating the performance of the organization at each time-step of the organization's life-span. On the one hand, organization cost defines the resources consumed by agents in terms of messages that are sent in the whole organization. On the other hand, benefits define the speed of each agent for completing its tasks. The organization performance is measured as the difference between the benefits and costs.

The cost measurement for carrying out the reorganization is specified for each agent in terms of a communication cost, which represents the reorganization costs. Furthermore, the impact of changing a relationship between a pair of agents is measured by taking into account the tasks that would or would not been assigned to other agents if a relationship is modified. This impact is associated to an increase or decrease of the tasks received by these other agents. However, a reflexive impact that measures the indirect costs and benefits caused to other agents is not represented. This impact should be related to how a relationship modification between a pair of agents can affect the agents that are not involved in the change, i.e. how these agents could allocate their own tasks if a relationship between different agents is modified. This issue is a consequence of the distributed reorganization process. Since reorganization

phases are carried out for each peer, the expected benefits and costs obtained are related to the information that is known by the pairs involved in the change. Therefore, if several reorganizations are carried out simultaneously between different pairs, the estimated costs and benefits might not be realistic since the structural topology may be different.

2.2.3.4 Evaluation

In this approach there is not an specific evaluation provided for measuring the proposal or the reorganization process. However, the performance of the future state of the organization that is achieved after reorganization is considered for future reorganizations, i.e., this evaluates which is the performance of the relationship after each modification.

The self-adaptive way of the TSE approach avoids a centralized implementation that could become a bottleneck. Nevertheless, since the process is not viewed from the organization perspective, the impact of each modification could be more difficult to measure than in centralized approaches, since every pair of agents does not have information about how other agents would adapt their relationship. As an example, if agents a_x and a_y form a relationship, several tasks could be allocated directly. However, other agents could also use this relationship to delegate tasks, causing an overload in a_x and a_y that would not be considered in the computation.

A reorganization viewed from the organization perspective would provide adaptation decisions that can be measured more consistently, since all the information required for reorganization is known by the whole organization. The impact associated to a relationship modification can be measured not only from the tasks that would not be allocated to other agents but also from the tasks that these other agents would allocate by taking into account this relationship modification. A distributed reorganization would provide more scalability, which would allow the applicability of the approach to problems that are composed by large agent organizations. However,

more efforts have to be done in order to keep the consistence of the system when several reorganizations are carried out simultaneously.

A disadvantage of this approach is that it is specifically constrained to structural reorganization and TSEs. Structural reorganization allow agents to reorganize their interactions in order to improve the utility of the whole organization. However, if other modifications were considered (e.g. changing the services provided by agents or changing the agent population in order to improve the performance), this utility might be improved since a wide range of alternatives would be considered. To the extent that we analyzed this approach, this model does not provide enough flexibility to incorporate reorganizations for different dimensions. One of the main advantages of this approach is the detailed computation of the reorganization consequences, at least for the agents involved in the relationship modification. This approach provides a specific detailed measurement of the benefits and some of the costs of the reorganization such as the cost of sending messages and changing relationships. Even though, the measurement is provided at the model level, the model is focused on specific applications related to this domain, making it difficult to extend these definitions to other applications.

2.2.4 Autonomic Electronic Institutions

Autonomic Electronic Institutions (AEIs) [18, 16] (Table 2.4) provide a paradigm for adapting the regulations of Electronic Institutions in order to accomplish institutional goals. In general, AEI involves the following elements: agents playing roles within scenes in a so-called performative structure, which defines the behavior of agents according to their role; goals, which are reached through interactions among agents; and norms, which specify the regulations of the system.

Table 2.3: Reorganization phases in TSEs

	Self-organization in TSEs
Monitoring	Proactive reorganization carried out in a distributed way by each pair of agents. The reorganization logic is predefined and cannot be changed, but the information that is monitored depends on the current relationships of each agent. A reactive reorganization is carried out when agents enter/exit the organization or when agents change their capabilities at predefined times.
Design	Each pair of agents designs each possible modification in their relationship. Structural adaptation: changes in the relationships.
Selection	Each pair of agents selects and implements the action that is better to their relationship. The utility function used for obtaining this alternative consider how the pair of agents involved in this change are affected in terms of benefits and costs. In addition, costs of sending messages and costs of changing relationships are also considered for measuring the reorganization costs.
Evaluation	The performance of the link after each modification is evaluated in a distributed way by each pair of agents.

2.2.4.1 Monitoring

The monitoring phase is implemented in a proactive way, in which several institutional agents are involved. These agents are able to detect situations such as a reorganization requirement due to a norm has been violated. The information that is required to be monitored is specified off-line but it is used on-line. The logic for reorganization is predefined at design time.

2.2.4.2 Design

The design phase is implemented in a centralized way by the institution itself. The changes that are considered are related to normative and performative structure adaptation. On the one hand, normative adaptation refers to changing the pre-conditions of a norm, its effects, or both. Since each norm is represented as a set of parameters, changing a norm is aimed at changing the values of these parameters. On the other hand, performative structure adaptation refers to changing the number of agents playing a role within each scene.

2.2.4.3 Selection

The institution explores the space of parameter values in order to find the configuration that provides the best degree of satistaction of institutional goals. After obtaining which are the values that provide the best degree of satisfaction of institutional goals, these values are set by the institution. A domain-dependent fitness function is defined to measure the degree of goal accomplishment; therefore, the objective of reorganization is to better accomplish these goals. As an example, in [16], a reorganization over a traffic scenario is proposed. In this example, goals are defined as a multi-attribute function that takes into account the number of accidents, the number of traffic offenses, and so on. Each time step, the AEI simulates different configurations for the penalties and institutional agents by using a learning model. In this scenario, norm adaptation is related to changing the penalties that are applied to cars that do not follow norms, while performative structure adaptation is related to changing the number of instutional agents in charge of detecting norm violations.

Costs for reorganization are not considered in AEIs. An agent of the AEI has an associated type of maintainance cost that limits the population of agents according to the benefits that these agents provide. However, this model does not incorporate mechanisms for measuring the impact (measured in terms of costs and benefits) of modifying a norm or the costs for carrying out the reorganization process. These mechanisms would provide more accurate simulations that consider not only the best performing configuration but also the best performing configuration by taking into account the costs of applying this configuration.

2.2.4.4 Evaluation

With regard to the evaluation phase, this reorganization approach do not incorporate mechanisms to measure how a reorganization has been carried out. However, in [17], they propose the use of case-based reasoning by the institution, in order to apply similar reorganization proposals under similar reorganization requirements. Even

Table 2.4: Reorganization phases in AEIs

	AEI
Monitoring	Proactive reorganization carried out in a distributed way by institutional agents. The information that is monitored is specified off-line but it is used on-line. The reorganization logic is predefined and cannot be changed.
Design	Centralized design carried out by the institution itself. Functional adaptation: changes in the number of agents playing a role. Normative adaptation: changes in the parameters associated to a norm.
Selection	The institution explores the space of values in order to find the configuration that maximizes the institution utility. Then, this configuration is set by the institution. The utility is measured as the benefits provided by all the agents.
Evaluation	Proposal evaluation that uses case-based reasoning for taking decisions under similar situations.

though this technique allows the prediction of the AEI with specific configurations, it does not measure how the reorganization process is carried out.

2.2.5 2-LAMA

The Two Level Assisted MAS Architecture (2-LAMA) [22, 21] (Table 2.5) is another approach that provides support for reorganization. This approach was first based on the AEI approach and then was extended to fit a more general model of organization. The organization in this approach is composed of a social structure that consists of a set of roles, groups, and the relationships among agents playing certain roles that belong to certain groups, social conventions that are expressed as interaction protocols and norms, and goals that describe the purpose of the organization.

Reorganization in the 2-LAMA approach is aimed at improving the accomplishment of the goals, for example, by modifying a norm.

2.2.5.1 Monitoring

Reorganization in the 2-LAMA approach can be reactive since several events occur (e.g. if an agent enters the organization), but it can also be proactive (e.g. if a norm is considered to be adapted in order to improve the accomplishment of the organization goals). Monitoring is carried out in a distributed way between assistant agents. The information required to be retrieved is specified off-line but it is used on-line without stopping the execution. The logic for reorganization is predefined at design time. Each assistant is in charge of managing the reorganization of a subset of agents. During the monitoring phase, each assistant perceives partial information about a cluster of agents and this information is shared with other assistants in order to take the decisions.

2.2.5.2 Design

After the monitoring phase, each assistant provides a reorganization proposal for each different component's related function based on the information available and the system goals. This corresponds to a distributed design phase.

Several adaptation functions are defined for updating the specific social structure and the norms of the organization. These adaptation functions evaluate the current organization in order to modify the specific components.

2.2.5.3 Selection

The assistants vote to select the reorganization, which is selected by agreement and implemented in a distributed way. Several criteria are used to select the specific changes in the 2-LAMA approach. As an example, in [21], assistant agents integrate two different methods based on heuristics and case-based reasoning. This approach considers reorganization costs (in time and/or resources) that should be taken into account in order to decide the reorganization frequency. Costs are computed by each

assistant and are classified into different categories such as the cost of collecting the information required, the cost associated to the time and resources that are required to compute the adaptation function, or the reorganization cost of transforming the organization into the adapted one. Costs define the minimal frequency required to keep the costs below the benefits that the reorganization generates. Otherwise, a higher frequency would cause a higher associated cost and, therefore, the reorganization may not be worth it.

However, these costs are not considered in the design nor in the selection phases. As we stated above, the driving force behind the reorganization is the accomplishment of the goals. Therefore, changes are introduced with the aim of inducing greater accomplishment of the current goals regardless of the costs, as long as these take into account the reorganization frequency. As an example, in peer-to-peer scenarios [21, 96], a social structure adaptation is carried out through modifications of relationships between peers with the aim of creating the optimal network composed by the paths with the shortest latencies. However, these modifications could have associated costs that would make it more profitable to adapt to a sub-optimal network that has an associated lower cost to be adapted to. Even though it is assumed that the reorganization frequency keeps costs below benefits, there is no support for achieving the reorganization with the highest tradeoff between costs and benefits.

2.2.5.4 Evaluation

With regard to the evaluation phase, the 2-LAMA approach does not provide mechanisms for evaluating the degree of success of the reorganization process. However, similar to the AEI approach, the 2-LAMA approach in [21] presents an example in which assistants use case-based reasoning for proposing a reorganization solution that is based on past experience. This kind of solution has been also extended to regulate the norms of dynamic systems [78].

One of the main advantages of this reorganization approach is the support for specify-

Table 2.5: Reorganization phases in 2-LAMA

	2-LAMA
Monitoring	Proactive reorganization carried out in a distributed way by assistant agents. Reactive reorganization is also supported when some event occurs. The information that is monitored is specified off-line but it is used on-line. The reorganization logic is predefined and cannot be changed.
Design	Distributed designs are proposed by assistant agents. Structural adaptation: relationships between agents. Normative adaptation: changes in the parameters associated to a norm.
Selection	Distributed decision between the assistants, which vote the selected design. The selected design is implemented by each assistant. This is aimed at obtaining a higher accomplishment of the goals (direct benefits caused by the reorganization). Reorganization costs: cost for transforming the organization into the new one. Computation costs: time and resources that are required to obtain the adaptation function.
Evaluation	Proposal evaluation in which assistants use case-based reasoning for taking decisions based on past experience.

ing the utilities for each individual component. Heterogeneous agents can be defined since different communication capacities are defined for each peer. Furthermore, each link between a pair of agents has its own associated communication capacity, which is determined by its bandwidth. Another advantage is the consideration of different dimensions of the organization to be adapted. However, support for jointly considering changes on different dimensions simultaneously should be required.

2.2.6 MACODO

MACODO (Middleware Architecture for COntext-driven Dynamic agent Organizations) [100] (Table 2.6) is a middleware that provides support for the management of organization adaptation. The organizational model used in this approach is composed by agents, capabilities, roles, and laws. Capabilities are viewed as agent abilities to perform tasks. A set of capabilities is required to play a role. Finally, laws describe the dynamic reorganization of organizations and define the consistence of the system.

2.2.6.1 Monitoring

Reorganization is reactively triggered by external events (e.g. when an agents stops playing a role) and changes in the environment (e.g. when the traffic state in the viewing range of an agent that collaborates in a traffic monitoring organization changes). Reorganization purposes are specified by means of two kinds of laws: intra-organization adaptation laws, which describe how agents can join and leave the organization (join and leave laws); and inter-organization adaptation laws, which describe the restructuring of organizations by merging and splitting organizations (merge and split laws).

A master controller agent is defined as being responsible for managing the dynamics of each organization in a centralized way. Each master controller enforces the laws that are related to the intra-organization adaptation of its organization (i.e. those that define when agents join or leave the organization). This describes the monitoring phase, which determines that a reorganization is required when a law is satisfied. The logic for reorganization is defined at design time. The information required by the master controller is specified off-line but it is used on-line.

Inter-organization adaptation requires information about more than one organization in order to enforce laws. In this kind of reorganization, masters of multiple organizations need to collaborate in a distributed monitoring phase. Each master exchanges a summary of the information regarding its organization with neighbouring masters. Similar to intra-organization adaptation, the monitoring phase is implemented through laws. As an example, when a merge law is satisfied in both organizations, a negotiation is initiated between the master of each organization involved in the merge in order to select the new master of the merged organization.

2.2.6.2 Design

Once an intra-organization adaptation is required, the master controller is in charge of designing the reorganization solution. In the case of inter-organization adaptation,

a master is selected, which carries out the actions required for the reorganization, completing the design phase. As we stated above, the changes that are supported are regarding open system support and structural adaptation.

2.2.6.3 Selection

The master controller that is responsible of the reorganization design, is also in charge of selecting and implementing this, which is finally carried out by updating the information regarding the organization.

In [100], a traffic example is presented to show the behavior of the middleware. In this example, master controllers decide to merge organizations when a traffic jam has arisen in the streets that they observe. Due to the environment change, agents decide to merge organizations because laws specify this condition. When the congestion starts to dissolve, agents split up the organization based on the split law. Therefore, we consider that the criteria is focused on accomplishing the organization goals when some change occurs.

In MACODO, a cost is related to communications, which is associated to the cost for merging and splitting organizations. This cost is used to evaluate the performance of the middleware. However, this cost is not taken into account to decide that a reorganization is required or to design the reorganization. As we stated above, reorganization is automatically caused when a law is triggered.

Event though a single master controller is in charge on carrying out the reorganization once it is required, we consider that phases can be also carried out in a distributed way because organizations can be viewed as sub-organizations since they can be merged. In this sense, master controllers are able to detect a reorganization requirement based on laws.

Table 2.6: Reorganization phases in MACODO

	MACODO
Monitoring	Reactive reorganization triggered by external events and changes in the environment. Centralized intra-organization adaptation carried out by a master controller, and distributed inter-organization adaptation carried out by several agent controllers. The information that is monitored is specified off-line but it is used on-line. The reorganization logic is predefined and cannot be changed.
Design	Depending on the reorganization type, the design can be centralized or distributed. Open system support and structural adaptation.
Selection	Centralized or distributed selection carried out by the master controller that is in charge of the design. The selected reorganization is implemented by this master controller as well. The selection is focused on fulfilling the organization goals when some event occurs and prevents the organization to accomplish its goals.
Evaluation	There is not any evaluation support implemented.

2.2.6.4 Evaluation

With regard to the evaluation phase, the MACODO approach does not provide mechanisms for incorporating information about the degree of success after a reorganization is carried out. Even though laws allow the organization to be adapted, these laws are defined at design time and cannot be modified depending on the information at runtime.

2.2.7 MAGIQUE

The approach implemented in the MAGIQUE platform [86, 71, 72] (Table 2.7) provides a reorganization mechanism that focus on two kinds of reorganization: individual adaptation and social adaptation. The agent society model defines an organization composed by agents, which are able to provide skills and can interact to each other according to the relationships structure, which defines the links between agents.

In this approach reorganization is viewed as a mechanism for improving the interaction between agents, which improves the system's performance. The underlying

reorganization mechanism is based on changing the relationships structure and the distribution of skills between agents in order to reduce the number of messages that are exchanged in the system, and the time necessary for processing a request. It is assumed that the overall system performance is an aggregation of the performance of individual agents. Therefore, if the performance of an agent increases, this does not negatively affect other agents of the system.

2.2.7.1 Monitoring

Monitoring in this approach is carried out in a distributed way since every agent is capable of deciding when a reorganization is required. In the current implementation, the monitoring strategy is reactive. Adaptation rules, which are specified by using thresholds, are used to trigger reorganization. These rules are predefined at design time. The information that is monitored is specified off-line but it is used on-line.

2.2.7.2 Design

After deciding that a reorganization is required, the design is autonomously carried out by each agent involved in the process. The elements that are considered to be changed are the relationships between agents (called acquaintances), the skills provided by agents (that can be mapped as the services that they provide), and the population of the system (by including new agents which can, in addition, learn specific skills). The decision of creating relationships and acquiring skills depends on policies that are specified by the system designer.

2.2.7.3 Selection

When an agent with the capabilities of designing a reorganization has carried out the design, this is automatically selected and implemented by using the API provided. The code mobility is used for learning skills.

The criteria used for selection is mainly focused on improving the benefits of the agent involved in the change, which is assumed to improve the overall performance of the whole organization as well. Therefore, indirect benefits and costs are not taken into account. In addition, there is not a model which provides support for defining the benefits that are associated to the reorganization, and this logic must be implemented by the user designer using his own metrics and techniques.

2.2.7.4 Evaluation

Regarding evaluation phase, there is not any kind of support for measuring how the reorganization has been carried out, except from those techniques that the agent designer implements by himself at agent level.

The main disadvantage of this approach is that the reorganization phases must be implemented by the system designer according to his own techniques and methodologies. As we stated above, a reactive monitoring is provided. However, reorganization could be considered to be carried out in a proactive way if the agent designer implements an underlying reasoning mechanism at agent level. In addition, if the reorganization is taken individually without considering indirect benefits and costs, this may cause different consequences as expected. In the examples given by the authors, the agent designer should consider how the addition of a link can affect other agents (these agents may take in turn some advantage or disadvantage that is not considered).

2.2.8 Reorganization in Agent-Organized Networks

Gaston & DesJardins [44] (Table 2.8) propose a reorganization approach for agent-organized networks (AONs) that is based on an agent team formation model. This model provides a dynamic environment in which agents form teams in a distribute way in order to accomplish the tasks that are received in the network. The society model defines an AON as a set of agents, which represent the nodes of the network,

Table 2.7: Reorganization phases in MAGIQUE

	MAGIQUE
Monitoring	Reactive reorganization triggered by rules. Distributed reorganization can be carried out by agents. The information that is monitored is specified off-line but it is used on-line without stopping the system. The reorganization logic is predefined and cannot be changed.
Design	Distributed design that can be carried out by any agent involved in the process. Structural adaptation: changes in the acquaintances; open system: changes in the population; behavioral adaptation: changes in the agent skills.
Selection	The agents involved in the design process select and implement the reorganization in a distributed way. The criteria used for this selection is to improve the direct benefits of the agents involved in the changes. The policy for this measurement must be implemented by the agent designer.
Evaluation	There is not any evaluation support implemented.

relationships between agents, which represent the adjacency matrix of the network, and skills, which are assigned to be provided by agents in order to perform tasks.

The reorganization that is considered in this approach is focused on allowing agents to modify their current relationships in order to improve the performance. Agents use local information in order to decide which links to delete and which to create.

2.2.8.1 Monitoring

Monitoring is carried out in a distribute way by any agent of the network. The monitoring strategy is proactive since each agent decides whether or not to adapt its links, according to its reasoning mechanism, which in the work [44] is based on a probability indicator. As agents remove their links and add new ones, we can consider that the information that is monitored can be specified on-line depending on the current set of links. In contrast, the reorganization logic (which is referred to the performance measurement), is predefined at design time.

2.2.8.2 Design

Once an agent has decided that a reorganization is required, this agent is in charge of designing the reorganization. In the AON this is referred at deciding which link is removed and which link is added, based on estimations of performance increase. Therefore, the design is also distributed. As we stated above, the elements that are allowed to be changed in this approach are the relationships between agents, which correspond to an structural adaptation.

2.2.8.3 Selection

After the design is proposed, this is automatically selected and implemented. The criteria used for this selection is based on the improvement of the performance, which is measured as the percentage of tasks for which teams successfully form. This improvement does not take into account how the link modification would affect other elements of the organization. As the same authors state, each agent has only a partial vision of the system and therefore, it is not possible to know all the information and how a change would influence. This model could be applied in scenarios in which an individual performance increase would be directly related to an organizational performance increase. The costs related to the application of the changes and to the reasoning process are not considered.

2.2.8.4 Evaluation

In this reorganization model there is not a support for evaluation. The same authors present in [20] an evaluation model for improving the team joining process. In this work, authors embedded agents in fixed network structures and focus on learning team joining policies. These policies improve the aggregate performance of the network. However, this learning model does not consider reorganization (modifications in the agent relationships). A similar learning approach would be interesting in or-

Table 2.8: Reorganization phases in AONs

	AONs
Monitoring	Proactive reorganization triggered by any agent of the network in a distributed way. The information that is monitored depends on the current relationships of the agent and it is also used on-line, while the reorganization logic is predefined and cannot be changed.
Design	Distributed reorganization carried out by the same agent that decided the reorganization requirement. Structural adaptation: changes in the links between agents.
Selection	The agents involved in the design process are the responsible of selecting and implementing this design in a distributed way. The criteria used for this selection is to improve the direct benefits.
Evaluation	There is not any evaluation support implemented.

der to select or predict the effect in the whole organization of adding and removing relationships, by defining policies for creating and deleting relationships.

One of the main problems of taking reorganization decisions that are based on local information is, as the authors claim, that local information provides a partial view of the system that may cause to take incorrect decisions, which cause an organizational performance decrease. In addition, if two agents decide to change their relationships simultaneosly, the reorganization benefit could not finally be as it was expected.

2.2.9 Comparison

Based on the analysis carried out, we summarize the main feautures of the analyzed approaches in Table 2.9. This table shows the different phases of the adaptation lifecycle based on the parameters defined in Section 2.1.

2.2.10 Other reorganization approaches

In this section, we mention other works that are focused on reorganization issues, but we do not provide an in-depth description of them. We only point out the most

relevant aspects. Some of these works only provide support to specific phases of the reorganization life-cycle. Other works are focused on reorganization issues over specific domains or applications and, thus, the contributions of these approaches would be difficult to extend to a general description that could be used in other domains. Finally, other works represent the basis of some of the works that are detailed in previous sections. We consider that these works also provide great contributions, to covering all the angles of the current state of the art of reorganization in agent societies.

In the work of [3], reorganization is viewed as a self-organizing mechanism that changes relationships between agents in order to reduce the total service time in a distributed task allocation domain. This approach uses a learning algorithm to optimize the agent policies based on how the agents interact with each other. In [49], a model is proposed for reorganization based on max-flow networks. In this approach the reorganization is focused on adding new agents and relationships to the organization. In the work of [97], an approach for a structural topology adaptation is proposed. This approach considers changes in the relationships between roles and the roles played by agents. Reorganization in this approach is carried out through predefined rules that cannot change during runtime. A similar work is proposed by [59]. This work focuses reorganization on organizational structure reorganization based on two primitives: spawning (creation of a new agent) and composition (merging several agents). Similar to the work of [21], this approach also considers the frequency of reorganizations. In [51], a system is proposed to provide support for changes in the interactions in a consumer-transporter domain. A diagnostic system is used to determine when an initial organization becomes potentially suboptimal. Then, changes in the interactions between consumers and transporters are proposed to overcome these insufficient resources. The reorganization is focused on minimizing the transport cost according to the load of transporters and the requirements of consumers. In this approach, several agents are involved in the selection of the reorganization. Another interesting work is the one proposed by [7]. In this work, different reorganization strategies for coping with different types of changes are presented. They

model changes that affect organizational structures and social relationships in a crisis management scenario.

Other works focus reorganization as being changes in the services provided or roles played by agents. As an example, in [87], a self-organization approach for a resource allocation environment is proposed. In this approach, the reorganization is carried out individually by each agent of the organization. The reallocation algorithm is implemented at the agent level, which takes allocation decisions based on predictions of the future resource utilization. Even though this approach is specifically focused on task allocation environments, an evaluation phase can be considered to be provided. After a task has been executed according to the reallocation algorithm, the prediction accuracy is evaluated and the history information is updated. Hoogendoorn et al. [50] propose a reorganization model that considers changes in the roles played by agents. In this approach, agents of a specific kind are in charge of a distributed monitoring of the organization goals. Reorganization is implemented by means of rules that are triggered when external circumstances cause that a requirement cannot be satisfied. When this occurs, a reorganization is needed. Similar to previous works, these rules are specified at design time. In the same context of role reallocation, the works [81, 80] present a technique for quantitatively comparing different role reallocations in order to decide which role allocation is the most beneficial. They use a reward-taking policy to assess the utility of the reorganization process by taking into account the benefits of a role reallocation process and the cost associated to it. However this technique is focused on a team decision problem; therefore, since the concept of organization is not considered, this technique could not be applied in domains that require changes in other elements of the organization. In the work [45], a reorganization model based on conventions is proposed. The objective of reorganization is to maximize the agent utilities and the organization utility. The utility of an agent depends on the roles it desires, the roles it has commited to, and the confidence that it has in its roles. In this approach, reorganization costs are defined. However, mechanisms for measuring these costs and the impact that changes produce in all the agents of the organization are not provided. In a similar line of research as

the previous approach, [90] proposes a model for reorganization that is focused on the role assignment to agents. These assignments try to maximize the utility of agents for performing each role. Reorganization in this approach is carried out by predefined norms that have associated specific actions for reorganization, such as modifying the organization in order to have more agents playing a specific role.

With regard to the regulations of the system, MLAW [24] is a middleware for supporting the dynamic modification of the system regulations during runtime. This framework supports changes in the permissions, prohibitions, and interaction protocols that govern the interactions among agents. Examples of changes are the exclusion of a permission that enables the entrance of an agent in a negotiation, the inclusion of a payment obligation, or changes in the negotiation protocol to include two new steps. The main drawback of this middleware is that changes are not reasoned by the system. These are specified and the centralized mechanism is in charge of law enforcement. The work of [94] presents a framework for allowing modification of the regulations of the system. Unlike the work of AEI (Section 2.2.4), this work supports the addition and deletion of norms and not only the modification of current norms. However, this support is not integrated in an organizational framework but is carried out by means of artifacts that control the norms of the system.

The work [48] is especially focused on the monitoring phase. In this paper, the authors propose an architecture for reorganization based on a replication mechanism. This architecture is composed of monitoring agents that are hierarchically organized and it provides support for a distributed observation mechanism. Host monitors exchange their local information in order to build global information. Then, agent monitors are in charge of executing the reorganization algorithms.

2.3 Discussion and Open Challenges

Given the analysis detailed in Section 2.2, there are some considerations that could be of great interest in future developments. We point out some of these issues below.

2.3.1 Monitoring

Detecting the reorganization requirement is a crucial phase for adaptive agent societies. There are some approaches that implement monitoring strategies as predefined rules that are triggered when some specified change occurs (OMACS, MACODO). Other approaches provide this phase through evaluations of the performance in time intervals during the organization life-cycle (TSEs), or also when a condition is accomplished such as in the 2-LAMA approach. Nevertheless, the rules that regulate the reorganization requirement are usually predefined at design time and cannot be changed while the organization is running. This forces designing systems in which the requirements for determining the reorganization must be known in advance, preventing the development of applications in which these requirements are not specifically known or which could even be different throughout the organization's life-span. Furthermore, the useful information required to be monitored in this phase is also usually specified off-line, before running the system.

It would be interesting for the next generation of adaptive agent societies to have support that allow the dynamic specification of the rules that trigger a reactive reorganization. As stated in [4], adaptive systems may cause monitoring requirements to also change. Thus, dynamic support that can adapt to these changes becomes essential in order to develop real adaptive agent societies. As an example, in the proposed workshop management system, a reactive reorganization can be required when an agent reviewer exits the system, which requires a reallocation of its assigned papers to other reviewers. However, this can be dependent (and also change) of execution factors. As an example, a restriction could be added at runtime which causes to not reallocate the specific paper if it has already two reviews. This support would provide more flexibility to dynamic systems, specially in scenarios in which is difficult to specify at design time the logic for reorganization.

Furthermore, this support should also consider changes in the relevant information that monitored. Thus, depending on the changing requirements of the system, the information required can change throughout the agent society's life-span. Static

Table 2.9: Comparison of reorganization approaches. (*Distr.=Distributed Centr.=Centralized*; *Proact.=Proactive*; *React.=Reactive*; *Predef.=Predefined*;)

		OMACS	Moise	TSE	AEI	2-LAMA	MACODO	MAGIQUE	AON
Monitoring	Implementation	Centr.	Distr.	Distr.	Distr.	Distr.	Centr./ Distr.	Distr.	Distr.
	Strategy	React.	Proact.	Proact./ React.	Proact.	Proact./ React.	React.	React.	Proact.
	Logic	Predef.	Predef.	Predef.	Predef.	Predef.	Predef.	Predef.	Predef.
	Information specificat.	Off-line	Off-line	On-line	Off-line	Off-line	Off-line	Off-line	On-line
	Information use	On-line	On-line	On-line	On-line	On-line	On-line	On-line	On-line
Design	Implementation	Centr.	Distr.	Distr.	Centr.	Distr.	Centr./ Distr.	Distr.	Distr.
	Open system		√				√	√	
	Emergence		√						
	Behavioural							√	
	Functional	√	√		√				
	Structural		√	√		√	√	√	√
	Normative		√		√	√			
Selection	Implementation	Centr.	Centr./ Distr.	Distr.	Centr.	Distr.	Centr./ Distr.	Distr.	Distr.
	Goal fulfillment	√	√	√		√	√		
	Direct Benefits	√	√	√	√	√		√	√
	Indirect Benefits								
	Direct Costs		√	√					
	Indirect Costs								
	Reorganization Costs			√		√			
	Computation Costs					√			
Evaluation	Implementation		Centr.	Distr.	Centr.	Distr.			
	Proposal				√	√			
	Reorganization process								
	Future state		√	√					

mechanisms that do not consider changes regarding which information needs to be monitored may result useful in small application domains with a priori well known organizational structures, but they would not be suitable for large-scale or complex systems. As the number of agents in the society and their complexity grows, much more information is exchanged between agents. Most of this information could be not useful at every moment of the execution and only contributes to considerably increase the traffic in the system, specifically in approaches in which a middleware or centralizing entity is the responsible of reorganization deliberation or implementation. Therefore, an adaptive approach should apply not only to the behavior and structure of the system, but also to the design of the monitoring system [85], specially when dealing with the management of complex systems over long periods of time.

2.3.2 Design

We have shown that several dimensions that we identified in Section 2.1.3 are separately covered by current approaches. This fact can be viewed through how this phase is implemented in each approach. Some of the current works focus on problems that approach reorganization in its functional dimension such as the OMACS approach, which changes the assignment of agents to roles. Other approaches deal with problems that require structural changes such as the works related with TSEs, AONs, or MAGIQUE. Other approaches such as AEI or 2-LAMA are specialized in changes in the regulations of the system. Note that there is not much open system support provided by current approaches, and what there is, is usually part of other adaptation support. As an example, the MACODO approach considers agents that can enter or leave the organization along with a support for inter-organizational adaptation.

All the surveyed approaches cover changes in different dimensions separately. However, we identified a lack of support in current approaches for behavior and specification adaptation. This means that the skills of agents are static and are not considered to evolve over time within the agent society context. In other words, agents are not

able to learn new capabilities or to degrade the utility of the capabilities that they offer. The MAGIQUE approach considers code mobility as the learning mechanism. However, this support must be provided at agent level by the agent designer. Furthermore, agent societies are not able to acquire new functionalities by means of the emergence of new roles (for example, as combinations of skills) or the deletion of old roles that are not effective.

As an example, in the workshop management system, this support would allow to develop systems in which reviewers are able to change the topics in which their are experts depending on the reviews. What is more, the dynamicity in the capabilities provided by agents may cause that new capabilities can emerge, e.g. a new topic which represents the interdisciplinary work of two trending topics.

As stated in [70], building systems with emergent behavior capabilities is important for increasing the robustness, autonomy, openness, and dynamism of the system. The application of some interesting principles adopted by human systems may facilitate the development of new models and mechanisms for agent societies that support the evolution of agent capabilities as well as the capabilities of the society. This could require using team learning or concurrent learning techniques to provide agents with the capabilities of discovering behaviours of other agents in the context of agent society [83].

Another remarkable consideration should be made regarding the support for a simultaneous reorganization in diferent dimensions. Most of the current approaches usually focus on changes in a specific dimension of the agent society such as the roles played by agents, the relationships between agents, or the norms of the system. However, few approaches consider several dimensions to be adapted at the same time. As an example, in the workshop management example, this would provide support to evaluate the tradeoff between adding new reviewers due to the high number of submissions, or reallocating all the submissions among the current reviewers. This support should consider the benefits and costs of each alternative.

Reorganization in agent societies usually involves changes that are focused on dif-

ferent dimensions such as organizing structures, coordination mechanisms, or work practices [82, 6]. Therefore, reorganization should consider different dimensions in order to increase the range of reorganization possibilities. In this line, the 2-LAMA approach provides an interesting view of reorganization since several dimensions can considered for reorganization. Even though reorganization in the work of [21] is only considered for structural and normative dimensions, the assistance layer provided by this approach would increase the number of dimensions considered for reorganization. However, reorganization in different dimensions should require a greater level of integration. In the current implementation of 2-LAMA, these changes are not considered simultaneously. Norm adaptation is considered at specific intervals of time, and structural adaptation is only considered each time an agent has completed a reception. Thus, the 2-LAMA approach would need to provide support for evaluating changes in different dimensions simultaneously and not at different moments with no dependence.

This implementation would require a greater level of integration between all the changes that can occur and their consequences. Thus, reorganization in several dimensions requires evaluating the interdependences of changes that could be applied simultaneously. As an example, a norm modification can influence a structural adaptation and vice-versa. The Moise approach can consider changes regarding different dimensions. However, the delegation of design and selection phases to individual agents limits this approach to methodologies provided by the designer.

Therefore, reorganization decisions that are dependent on several dimensions would require more complex deliberation processes that assess the suitability of the reorganization. Future reorganization approaches should provide a higher level of integration between changes in different dimensions and their interdependence.

2.3.3 Selection

With regard to the criteria used to estimate the reorganization suitability, we observe that costs related to reorganization have not usually been taken into account to take reorganization decisions. As stated in [36], a reorganization process should provide some kind of increase in utility. However, as far as we are concerned, this utility should take into account not only the gain in utility but also the costs of carrying out the reorganization. Some approaches do consider certain costs; however, these costs are not accurately estimated. One of the approaches that provides a measurement of several implications of a change is the approach of TSEs. This approach measures the impact of changing a relationship between a pair of agents not only in terms of benefits but also in terms of some costs. Thus, this approach evaluates how the load of agents can be changed depending on the relationship modification. However, as we stated in Section 2.2.3, since the reorganization approach is completely carried out in a distributed way by each pair of agents, the estimation of the implications that a change could cause in the rest of the population of agents is difficult to measure. Only the tasks related to the agents involved in the change that would or would not be delivered to other agents are considered. In the 2-LAMA framework several costs are also considered for reorganization. However, these costs are only used to define the reorganization frequency. This approach would require considering the reorganization impact for selecting the reorganization.

As we have shown, current approaches usually assume that a relationship modification between a pair of agents or a change in the role played by an agent do not have implications in the rest of the population and can be carried out without requiring additional costs (time, resources, etc.) [71, 31, 16]. However, as in human societies, not every change in an agent society has the same implications in terms of costs or has the same impact in the whole society.

As an example, in the workshop management system, this is referred as considering not only the benefits of adding reviewers into the society, or reallocating the papers assigned, or creating a new role. This refers to also consider how changes affect

other elements of the society in terms of benefits and costs. As an example, the reallocation of a paper to the most confident reviewer of this topic may cause several reallocations that are negative for the society. All of these positive and negative impacts along with other costs that are associated to how much time requires to carry out each reallocation could be represented by cost-aware approaches.

Therefore, it would be interesting for future approaches to provide mechanisms that use metrics for an accurate evaluation of the reorganization implications (positive and negative). These evaluations should consider not only the agents involved in the change but also how a change can influence the performance of the rest of the agents of the society, as well as the costs associated to carrying out the reorganization process itself.

2.3.4 Evaluation

The reorganization phase that has probably received the least attention by researchers is the evaluation phase. It is a general assumption by theorists of human societies that these societies learn from experience by means of the changes that take place in their history [69, 63, 47]. Similarly, the importance of evaluating the reorganization process seems to be clear in agent societies. Feedback provides important information about whether changes have been implemented as intended. This is important for agent societies to be able to develop responses that reuse old solutions to problems [61, 47].

Current approaches do not invest very much effort in evaluating the reorganization process due to the difficulties associated to solving this problem. It is assumed that when the reorganization is carried out, the process is implemented as expected (without any setback or indirect consequences). Indeed, many approaches such as OMACS or MACODO concentrate on applying a solution to a problem, which is triggered by predefined rules, assuming that the reorganization solves the problem for which it was proposed and no revision of the mechanism that estimates the solution is required at

execution time. Few approaches such as 2-LAMA consider that past information related to the reorganization decisions to be considered in the future. This approach uses case-based reasoning for deciding similar reorganization solutions to similar problems that were solved in the past. Nevertheless, an effective evaluation of the reorganization process is not provided by any approach. This effective evaluation should not only be useful to trigger the reorganization, but it should also adjust the parameters that estimate the reorganization consequences (in terms of benefits and costs) based on how the reorganization has been applied.

Information provided by the historian agent proposed by Moise in [54] is an interesting approach to determine which design solution is more suitable based on the performance of past design solutions. Although solutions to provide an effective evaluation may be dependents on the specific domain, more efforts are needed to better understand the reorganization life-cycle in order to incorporate information regarding the reorganization application in future reorganization considerations.

Related to the example of the workshop management system, a stronger evaluation support would determine whether or not the reallocation of a paper to another reviewer caused the benefits and costs as expected. If not, the system should learn in order to improve the prediction accuracy for future reorganizations or could also adapt other elements (e.g. the society learns that when some paper reallocation is required, the notification deadline must be also extended).

In addition, the evaluation should also provide a feedback regarding the costs (time, resources, etc.) required to carry out the reorganization. Learning can also be applied to other dimensions of the evaluation. It can be difficult to predict the impact of a change in advance. However, information regarding how the estimation of similar changes was in the past could be useful to improve this prediction accuracy.

2.4 Conclusions

In this chapter, we have analyzed some of the most relevant contributions related to reorganization in agent societies, detailing the reorganization models that were used as well as how this support is provided and implemented. The specification of each phase of the life-cycle of the reorganization process presented in Section 2.1 allows us to compare the different approaches according to what they provide and what they do not provide. We have analyzed in detail the most relevant approaches (OMACS, Moise, TSE, AEI, 2-LAMA, MACODO, MAGIQUE, and AON), offering a general overview of the different phases of the reorganization life-cycle.

The criteria that we used to define the dimensions in Section 2.1 was based on properties that emerge from human organizations as well as properties that are defined by relevant works of the agents research area. More detailed properties could be found in specific phases, such as whether the cost is represented in a time domain or in a resource domain. Nevertheless, the objective of this classification was to provide a set of general parameters and dimensions that can be used to compare a wide range of approaches.

The goal of this review chapter is to provide a detailed analysis of the most relevant existing approaches, in order to show the advantages and limitations of each one. In addition, the open challenges that have been pointed out in Section 2.3 can help system designers in order to develop new frameworks and reorganization models to overcome the main limitations of current approaches. Some of these challenges can be highly dependent on the domain while other could be addressed to provide general solutions:

- New monitoring approaches to support dynamic specification of the rules or methods that trigger a reorganization process as well as the dynamic specification of the information required to be analyzed during the monitoring phase.

- New models that allow agents to dynamically change their capabilities within

the context of the agent society. This support would be extended to emergent social behavior.

- New design techniques that consider several dimensions to be adapted simultaneously in the reorganization process. This issue would require estimating the interdependence of changes in different dimensions.
- New models that estimate a detailed measurement of the impact that a change has on the rest of the elements of the agent society. This impact should be measured not only in terms of benefits but also in term of costs.
- New evaluation support that measures the degree of success of a reorganization once it has been applied. This evaluation should consider how the reorganization has been carried out and how it improves the accuracy of the estimation.

In order to resolve some of these open issues, future research works are encouraged to invest their effort in this area of reorganization in agent societies. In the following chapters we present our reorganization approach that is focused on dealing with some of these open challenges, specially those related to the more essential support, which is viewed as the reorganization model support.

Part III

Reorganization Model

CHAPTER

3 Organization Transition Model

3.1 Organization 66
3.2 Organization transition 67
3.3 Organization Transition Impact 71
3.4 Multi-dimensional Transitions 74
3.5 Conclusions 81

In this chapter we present the reorganization model that provides support for dynamic organizations, which considers changes in different dimensions and provides an accurate measurement of the adaptation consequences. This reorganization model is based on the concept of organization transitions [31] which allow us to relate two different organizations at different moments. First, we define the organization model, which represents an organization at a given moment; the organization transition, which relates two organizations at different moments; and the computation of the impact related a given transition. Then, we show how this impact is represented in a multi-dimensional transition approach.

3.1 Organization

Models for agent societies allow to represent the elements that make up the society as well as the interactions among these elements. Several approaches can be found in the literature for modeling agent societies depending on the requirements of the applications. Current models have been compared and reviewed by works such [95, 33, 10].

Though several approaches can be used to model agent societies, we use an adaptation of the organization proposed model by Esparcia and Argente [40] since we found it to be sufficient for the requirements of the model proposed. We require a model that represents the agents of the society, the relationships between these agents, the services that these agents provide, and the roles that agents play according to these services. Furthermore, we require the model to provide the dynamism necessary for representing different states of the society at different moments. Thus, we adapt this organization model to group the elements of the organization according to these requirements.

Definition 1 (Organization). An organization at a specific moment t is defined as a tuple $O^t = \langle O^t_O, O^t_R \rangle$ where:

O^t_O stands for *Organizational Objects* and represents individual objects of the organization.

It is defined as $O^t_O = \{R^t, S^t, A^t\}$, where:

- R^t represents the set of roles contained in the organization at a specific moment t.
- S^t represents the services that the organization is offering at a specific moment t.
- A^t represents the population of agents at a specific moment t.

O_R^t stands for *Organizational Relationships* and represents relationships of the organization by means of a link between the objects.

It is defined as $O_R^t = \{offers^t, provides^t, plays^t, acquaintance^t\}$, where:

- $offers^t = \{(r, s) \in R^t \times S^t\}$ represents relationships between roles and services, where (r, s) represents that the role r offers the service s at moment t.
- $provides^t = \{(a, s) \in A^t \times S^t\}$ represents relationships between agents and services, where (a, s) represents that the agent a provides the service s at moment t.
- $plays^t = \{(a, r) \in A^t \times R^t\}$ represents relationships between agents and roles, where (a, r) represents that the agent a plays the role r at moment t.
- $acquaintance^t = \{(a, b) \in A^t \times A^t\}$ represents the relationships between a pair of agents, where (a, b) represents that the agents a and b are connected by an acquaintance relationship at moment t. These relationships define the structural topology of the organization.

Given an organization O^t at a specific moment t, an agent a is able to play a role r at time t, if a provides all the services s that r offers at time t:

$$\forall (a, r) \in plays^t \mid (r, s) \in offers^t \rightarrow (a, s) \in provides^t$$

3.2 Organization transition

The concept of *organization transition* allows us to relate two different organizations at different moments, current (c) and final (f). This concept was first introduced by Matson and DeLoach [76]. In this work, an organization transition is the mechanism by which an organization is reorganized into a new one.

This mechanism is based on individual changes that are applied to objects and relationships of O^c in order to obtain the objects and relationships of O^f. We can divide the organization transition into two transitions. On the one hand, the *objects transition function* (γ) is the mechanism that supports the transition of the objects of O^c into the objects of O^f, that is, individual changes that can applied to the objects of O^c for obtaining the objects of O^f. On the other hand, the *relationships transition function* (δ) is the mechanism that supports the transition of the relationships of O^c into the relationships of O^f, that is, individual changes that can applied to the relationships of O^c for obtaining the relationships of O^f. The application of both transition functions allows an organization transition from O^c to O^f.

Definition 2 (Events). An *event* (ε) defines each individual change that can be applied to an object or to a relationship during the organization transition. It is expressed as a function in terms of addition or deletion of individual objects or relationships. Given an organization $O^t = \langle O^t_O, O^t_R \rangle$, where $O^t_O = \{R^t, S^t, A^t\}$ and $O^t_R = \{offers^t, provides^t, plays^t, acquaintance^t\}$, an addition event causes the specific object or relationship to be added to the specific set of O^t (Table 3.1).

Event	Precondition	Effect
$add_role(r)$	$r \notin R^t$	$R^t \rightarrow \{R^t \cup r\}$
$add_service(s)$	$s \notin S^t$	$S^t \rightarrow \{S^t \cup s\}$
$add_agent(a)$	$a \notin A^t$	$A^t \rightarrow \{A^t \cup a\}$
$add_offers((r,s))$	$r \in R^t \wedge s \in S^t$ $\wedge (r,s) \notin offers^t$	$offers^t \rightarrow \{offers^t \cup (r,s)\}$
$add_provides((a,s))$	$a \in A^t \wedge s \in S^t$ $\wedge (a,s) \notin provides^t$	$provides^t \rightarrow \{provides^t \cup (a,s)\}$
$add_plays((a,r))$	$a \in A^t \wedge r \in R^t$ $\wedge (a,r) \notin plays^t$	$plays^t \rightarrow \{plays^t \cup (a,r)\}$
$add_aquaintance((a,b))$	$a \in A^t \wedge b \in A^t \wedge$ $(a,a) \notin acquaintance^t$	$acquaintance^t \rightarrow$ $\{acquaintance^t \cup (a,b)\}$

Table 3.1: Addition events

Similar to addition events, a deletion event applied to an object or to a relationship causes it to be deleted from the specific set (Table 3.2).

Event	Precondition	Effect
$delete_role(r)$	$r \in R^t$	$R^t \rightarrow \{R^t - r\}$
$delete_service(s)$	$s \in S^t$	$S^t \rightarrow \{S^t - s\}$
$delete_agent(a)$	$a \in A^t$	$A^t \rightarrow \{A^t - a\}$
$delete_offers((r,s))$	$(r,s) \in offers^t$	$offers^t \rightarrow \{offers^t - (r,s)\}$
$delete_provides((a,s))$	$(a,s) \in provides^t$	$provides^t \rightarrow \{provides^t - (a,s)\}$
$delete_plays((a,r))$	$(a,r) \in plays^t$	$plays^t \rightarrow \{plays^t - (a,r)\}$
$delete_acquaintance((a,b))$	$(a,b) \in acquaintance^t$	$acquaintance^t \rightarrow$ $\{acquaintance^t - (a,b)\}$

Table 3.2: Deletion events

Definition 3 (Set of events). Given two organizations, $O^c = \{O^c_O, O^c_R\}$ and $O^f = \{O^f_O, O^f_R\}$, a set of organizational object events $\tau_{O_O} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ comprises events that, cause a transition to O^f_O, when all of them are applied to O^c_O. A set of organizational relationship events $\tau_{O_R} = \{\varepsilon'_1, \varepsilon'_2, \ldots, \varepsilon'_n\}$ comprises events that, cause a transition to O^f_R, when all are applied to O^c_R.

Thus, we define the *set of events* τ that allows a transition from O^c to O^f as a set of events regarding both objects and relationships that, cause a transition to O^f, when all of them are applied to O^c:

$$\tau = \{\tau_{O_O} \cup \tau_{O_R}\}$$

Definition 4 (Dependency of events). An event $\varepsilon \in \tau$ is *dependent* on another event $\varepsilon' \in \tau$ if, in order for ε to be applied, ε' must first be applied. The function $dep(\varepsilon, \tau)$ obtains the events of τ that ε is dependent on. As an example, if there are two different events $\varepsilon = add_plays((a,r))$ and $\varepsilon' = add_role(r)$ in the same set of events τ , we say that $\varepsilon' \in dep(\varepsilon, \tau)$ because ε requires that ε' be applied before ε. In contrast, two events $\varepsilon = add_agent(a)$ and $\varepsilon' = add_role(r)$ are *independent* if they can be applied simultaneously during the transition process, that is, $\varepsilon' \notin dep(\varepsilon, \tau) \wedge \varepsilon \notin dep(\varepsilon', \tau)$. The dependency between events defines which ones could be applied simultaneously during the transition process and which ones must

be applied sequentially. The dependency of events that refers to the same objects provides organization consistency during the transition process, such as:

$$add_role(r) \in dep(add_plays(a, r)), \tau)$$

In this example, we need the role r to be added to the organization before the agent a plays it. We can also define dependencies of events for specifying restrictions depending on the domain. As an example, if agent a requires providing service s_1 before providing service s_2, this dependency can be expressed as:

$$add_provides((a, s_1)) \in dep(add_provides((a, s_2)), \tau)$$

$$delete_provides((a, s_2)) \in dep(delete_provides((a, s_1)), \tau)$$

Definition 5 (Subsets of events). Given a set of events τ that allows a transition from O^c to O^f, if an event $\varepsilon_i \in \tau$ is dependent on another event $\varepsilon_j \in \tau$, such that $\varepsilon_j \in dep(\varepsilon_i, \tau)$, then τ must be split into two subsets $\tau = \{\tau_1, \tau_2\}$, where $\varepsilon_j \in \tau_1$ and $\varepsilon_i \in \tau_2$. The application of the first subset τ_1 allows a transition from O^c to an intermediate organization O^{int}, and the application of the second subset τ_2 allows a transition from O^{int} to O^f. Therefore, a set of events τ can be represented as a sequence of subsets of events $\tau_1, \tau_2, \ldots, \tau_n$ ordered by a dependency order, i.e., at least one event of τ_{i+1} is dependent on one event of τ_i whatever i is. Therefore, each τ_i groups events that are independent:

$$\forall \tau_i \in \tau, \forall \varepsilon, \varepsilon' \in \tau_i \rightarrow \varepsilon' \notin dep(\varepsilon, \tau_i) \wedge \varepsilon \notin dep(\varepsilon', \tau_i)$$

Given two subsets of events $\tau_i, \tau_j \subset \tau$ such that $i < j$, there is no event in τ_i that is dependent on an event of τ_j:

$$\forall \varepsilon, \varepsilon' \mid \varepsilon \in \tau_i \wedge \varepsilon' \in \tau_j \rightarrow \varepsilon' \notin dep(\varepsilon, \tau_j)$$

Furthermore, at least one event of τ_j must be dependent on one event of τ_i:

$$\exists \varepsilon \in \tau_i, \exists \varepsilon' \in \tau_j \mid \varepsilon \in dep(\varepsilon', \tau_i)$$

Definition 6 (Transition path). When a set of events τ_i is applied to an organization O^c to cause a transition to O^f, we say that both organizations are connected by a *direct transition*. If a sequence of subsets $\tau_1, \tau_2, \ldots, \tau_n$ is applied to transition from O^c to O^f, we say that both organizations are connected by an *indirect transition*. In this case, the application of each $\tau_i \subset \tau$ causes a direct transition to an intermediate organization. The sequence of organizations that is reached in the transition between O^c and O^f represents a *transition path* between both organizations.

3.3 Organization Transition Impact

In order to calculate the organization with the highest potential for improvement in utility based on the transition cost for several changes, we define the concept of Organization Transition Impact (OTI). The OTI is a measurement of the effects of an organization transition in terms of organization utility based on the costs for carrying out this transition. Computing the OTI becomes essential in order to empirically specify the value of this transition in terms of time consumption, money, resources, and so on.

The application of the set of events τ associated to an organization transition provides us with information regarding what changes must be carried out in order to fulfil the transition. Each event $\varepsilon \in \tau$ has an associated impact $i(\varepsilon)$ if ε is applied. This impact represents the costs/benefits that the application of this event causes in the organization. This impact shows the effect of this event in the components involved in the change and also how other components are affected by this event. Moreover, the impact shows the cost for carrying out the application of the event.

For any set of events τ that allows a transition from the current organization O^c to a

future organization O^f, we define the impact that is associated to the organizational objects $i(\tau_{O_O})$ as the impact of applying all the events associated to objects. This impact is computed as the aggregated impact of the events that allow a transition from O_O^c to O_O^f and entails operations of addition and deletion of roles, services, and agents:

$$i(\tau_{O_O}) = \sum_{\varepsilon \in \tau_{O_O}} i(\varepsilon)$$

Similarly, we define $i(\tau O_R)$ as the impact of applying all the events associated to relationships. This impact allows a transition from O_R^c to O_R^f and refers to addition and deletion events of $offers$, $provides$, $plays$, and $acquaintance$ relationships:

$$i(\tau_{O_R}) = \sum_{\varepsilon \in \tau_{O_R}} i(\varepsilon)$$

Finally, we can compute the OTI as the impact caused by applying all the events τ that cause the transition regarding both objects and relationships:

$$OTI(\tau) = i(\tau_{O_O}) + i(\tau_{O_R}) = \sum_{\varepsilon \in \tau} i(\varepsilon)$$

If τ is composed by a sequence of ordered subsets $\tau_1, \ldots, \tau_n$, the OTI is represented as the aggregation of the impact of all the subsets:

$$OTI(\tau) = \sum_{\tau_i \in \tau} OTI(\tau_i)$$

Each organization transition focused on a specific dimension, provides the future organization O^f that could be transitioned to, which minimizes the OTI. In the following chapter we define each organization transition according to the events that are associated to each one.

3.3.1 Organization Transition Impact and Organization Utility

The OTI reflects the organization utility improve according to the organization transition costs. The utility function of the organization $U(O^c)$ at the current moment c can be represented as a multi-attributed utility based on approaches from Decision Theory [77]. This multi-attributed utility can effectively deal with both qualitative and quantitative factors in multiple criteria and uncertain decision environments and is represented as:

$$U(O^c) = \sum_{i=1}^{n} k_i \times U_i(O^c)$$

Where each k_i represents relative weights associated to the attribute i. These weights represent a decision maker's judgments on the relative importance or preference of the attributes. The utility $U_i(O^c)$ represents the utility of the organization for attribute i.

An organization transition causes that a future organization O^f is reached, which has an associated utility of $U(O^f)$. In addition, the transition has an associated cost for being carried out that is defined as $C(O^c, O^f)$. An alternative for computing the OTI associated to a transition between two organizations is to evaluate this OTI as the benefits reached by the transition according to the costs required to carry out the transition. Therefore, the OTI for an organization transition from O^c to O^f can be defined by the following equation:

$$OTI(O^c, O^f) = U(O^c) - U(O^f) + C(O^c, O^f)$$

The difference between both utilities represents the OTI on the utility of the organization while the $C(O^c, O^n)$ represents the OTI on the costs for achieving the future organization. According to this definition, a negative impact represents that the transition would be beneficial for the organization.

3.4 Multi-dimensional Transitions

Most of the existing approaches focus the adaptation on specific dimensions of the organization. Works such as [65, 71, 97, 59] propose adaptation models that are based on changes in the agent relationships in order to obtain better performance; other works such as [16] propose an adaptation in terms of norms by changing the regulations of the system; and other works are focused on changes in the roles played by the agents [81, 50, 75, 90]. Also, few models provide mechanisms for measuring the impact of the adaptation in the whole organization. Most approaches consider an adaptation decision that is focused on the increase of the utility [75, 16]. However, the costs associated with carrying out the adaptation process and the costs/benefits affecting other agents as side effects of the adaptation have not been widely taken into account.

In the following sections, we define how transitions in three different dimensions can be represented. These transitions consider changes in the services provided and roles played by agents (role reallocation transition), changes in the structural typology (acquaintance transition), and changes in the population of agents (agent population transition).

3.4.1 Role Reallocation Transition

The organizational relationships $provides$ and $plays$ represent the services that are provided by each agent and the roles that are played by each agent, respectively, at a specific moment. Nevertheless, for a given cost, an agent can provide other services and can play different roles that it is not currently playing. The organization transition that focuses on these changes is called the *role reallocation transition*.

Given the specification of the organizational objects of the final organization that is to be achieved O_O^f, and the organizational relationships $offers^f$ and $acquaintance^f$ that are to be achieved, some agents could be reallocated to provide other services

and to play other roles that they were not playing in O^c. A role reallocation transition entails the application of a specific set of events τ_R composed by $provides$ and $plays$ relationships, which transforms the $provides^c$ and $plays^c$ relationships into $provides^f$ and $plays^f$, respectively. Each one of these role reallocations determines a different O^f that could be transitioned to by applying a set of events τ_R with an associated $OTI(\tau_R)$.

The OTI related to the role reallocation transition measures how costly it is for agents to acquire the services to play a specific role, to start playing this role, to stop playing a role that is currently being played by an agent, and to stop providing the services required for this last role. This impact also measures how beneficial it is for the agents involved in the role reallocation and for the organization to have these agents change their roles.

To calculate the OTI of a role reallocation transition, we need to estimate the impact related to the events required for each agent that is reallocated to play a new role. We define the impact of agent a_x for acquiring the services offered by a new role r_n that are not already provided by the agent as:

$$I_A(a_x, r_n) = \sum i(add_provides((a_x, s_n)))$$

for every service s_n that $(r_n, s_n) \in offers^f \wedge (a_x, s_n) \notin provides^c$. Once the agent a_x provides these services, it can start playing the role r_n for an impact of $i(add_plays((a_x, r_n)))$. Thus, the whole impact of a_x for start playing r_n is defined as:

$$I_S(a_x, r_n) = I_A(a_x, r_n) + i(add_plays((a_x, r_n))) \tag{3.1}$$

The impact of agent a_x to stop playing the current role r_c is defined as $i(delete_plays((a_x, r_c)))$. Once a_x does not play this role, it can stop providing the services required to play r_c

that are no longer required for playing other roles in O^f for an impact of:

$$I_D(a_x, r_c) = \sum i(delete_provides((a_x, s_c)))$$

for every service s_c that $(a_x, s_c) \in provides^c \wedge \nexists r_d \mid (r_d, s_c) \in offers^f \wedge (a_x, r_d) \in plays^f$. Thus, the whole impact of agent a_x to stop playing a current role r_c is defined as:

$$I_P(a_x, r_c) = i(delete_plays((a_x, r_c))) + I_D(a_x, r_c) \tag{3.2}$$

Therefore, we define the impact of role reallocation for agent a_x from role r_c to role r_n by taking into account the impact related to stop playing r_c in order to play r_n:

$$I_R(a_x, r_c, r_n) = I_S(a_x, r_n) + I_P(a_x, r_c) \tag{3.3}$$

Thus, the OTI associated to the set of events τ_R that causes a role reallocation transition from O^c to O^f can be written as:

$$OTI(\tau_R) = \sum_{a_x \in A} I_R(a_x, r_c, r_n)$$

where $(a_x, r_c) \in plays^c \wedge (a_x, r_n) \in plays^f$.

Let Θ_R denote the set of all the possible sets of events τ_R that define a different role reallocation transition from O^c and obtains a different O^f. The challenge of the role reallocation transition is to find the specific set of events $\widehat{\tau}_R$ that minimizes the role reallocation transition impact:
The application of the role reallocation transition algorithm calculates the set of events of the minimal impact $\widehat{\tau}_R$ and the final organization O^f that is to be transi-

$$OTI(\widehat{\tau}_R) = \underset{\tau_R \in \Theta_R}{\operatorname{argmin}}\, OTI(\tau_R) \tag{3.4}$$

tioned to by applying this set of events. The implementation of the algorithm (line 5 of Algorithm 1) considers both, the role swap between agents and the change in the number of agents that play a specific role.

The transition path of the minimal impact for the role reallocation transition, defines a transition from O^c to O^f, which has associated the set of $provides$ and $plays$ events of the minimal impact $I(\widehat{\tau}_R)$.

3.4.2 Acquaintance Transition

Organizational relationships represented in $acquaintance$ define the structural topology of the organization, by defining which agents are related to each other at a specific moment. Acquaintances between a pair of agents can be modified at a given cost, and this may change the performance not only of the agents involved in the relationship but also the utility of the whole organization. The organization transition that is focused on changes regarding acquaintances between agents is called the *acquaintance transition*.

Given the specification of the organizational objects of the final organization that is to be achieved, O_O^f, and the organizational relationships $offers^f$, $provides^f$, and $plays^f$ that are to be achieved, some acquaintances can be created between a pair of agents that were not related in O^c, and some acquaintances between agents that were related in O^c can be deleted. An acquaintance transition entails the application of a specific set of events τ_A composed by $acquaintance$ relationships , which transforms $acquaintance^c$ into $acquaintance^f$. Each one of these specific $acquaintance^f$ relationships defines a future organization O^f, which represents a specific structural

topology and can be achieved by applying a specific set of events τ_A with an associated $OTI(\tau_A)$.

The OTI related to the acquaintance transition measures how costly it is for a pair of agents to create an acquaintance relationship between them or to delete an existing relationship, and how these modifications affect the utility of the organization.

The impact of adding an acquaintance relationship between a_x and a_z is defined as $i(add_acquaintance((a_x, a_z)))$. This represents the cost for relating these agents from this moment on and how this relationship affects the utility of the organization. The impact of deleting an existing acquaintance relationship between a pair of agents a_x and a_z is defined as $i(delete_acquaintance((a_x, a_z)))$. This represents the cost for these agents to no longer be related and how this affects the utility of the organization.

Let Θ_A denote the set of all the possible sets of events τ_A that define a different acquaintance transition from O^c and obtains a different O^f. The challenge of the acquaintance transition is to find the specific set of events $\widehat{\tau}_A$ that minimizes the acquaintance transition impact:

$$OTI(\widehat{\tau}_A) = \underset{\tau_A \in \Theta_A}{\operatorname{argmin}}\, OTI(\tau_A) \tag{3.5}$$

The application of the acquaintance transition algorithm calculates the set of events of the minimal impact $\widehat{\tau}_A$ and the final organization O^f that is to be transitioned to by applying this set of events. Each execution of the algorithm (line 6 of Algorithm 1) takes into account the addition or deletion of a single relationship in the organization but several iterations can be carried out due to the loop (lines 4-9 of Algorithm 1).

The transition path of the minimal impact for the acquaintance transition, defines a transition from O^c to O^f, which has associated the set of $acquaintance$ events of the minimal impact $I(\widehat{\tau}_A)$.

3.4.3 Agent Population Transition

Organizational objects represented in $agents$ define the population of agents at a specific moment. Nevertheless, the population of agents can be changed at a given cost, which may imply that the number of agents playing a specific role or the number of acquaintance relationships of some agents also changes. The organization transition that focuses on these changes is called the *agent population transition*.

Given the specification of the organizational objects $roles^f$ and $services^f$ of the final organization that are to be achieved, and the organizational relationships $offers^f$ that are to be achieved, some agents that were not in O^c can be added into the organization, and some agents that were in O^c can be deleted from the organization. An agent population transition modifies the agent population $agents^c$, and this may also cause modifications in the $provides$, $plays$, and $acquaintance$ relationships. Thus, an agent population transition entails the application of a set of events τ_P, which causes the modification of $agents^c$, $provides^c$, $plays^c$, and $acquaintances^c$ into $agents^f$, $provides^f$, $plays^f$, and $acquaintances^f$, respectively. Each one of these specific $agents^f$, $provides^f$, $plays^f$, and $acquaintances^f$ defines a future organization O^f, which represents a different configuration and can be achieved by applying a specific set of events τ_P with an associated $OTI(\tau_P)$.

The OTI related to the agent population transition measures how costly it is to add or delete agents of the organization and how this affect to the utility of the organization.

The impact of adding an agent a_x into the organization is defined as $i(add_agent(a_x))$. Moreover, acquaintance relationships between this agent and a set of agents $A' \subset A$ may be created with an associated impact that is defined as:

$$I_A(a_x) = \sum_{(i) \in A'} i(add_aquaintance((a_x, i)))$$

Finally, this agent may also provide some services and play a specific role r_n based on the services provided. Thus, we define the impact of agent a_x start playing role r_n

as $I_S(a_x, r_n)$ from Eq. 3.1.

We represent the whole impact of adding an agent a_x as:

$$I_S(a_x) = i(add_agent(a_x)) + I_A(a_x) + I_S(a_x, r_n) \tag{3.6}$$

The impact of deleting an agent a_x from the organization is represented as $i(delete_agent(a_x))$. Moreover, acquaintances that involve this agent must be deleted with an impact defined as:

$$I_D(a_x) = \sum i(delete_aquaintance((i,j)))$$

for all $(i,j) \in acquaintance^i \mid i = a_x \vee j = a_x$.

Finally, deleting a_x from the organization has an impact of stopping a_x from playing the role r_c that it is playing as well as the impact of stopping a_x from providing its services. This impact is represented as $I_P(a_x, r_c)$, from Eq. 3.2. We represent the impact of deleting an agent a_x as:

$$I_P(a_x) = i(delete_agent(a_x)) + I_D(a_x) + I_P(a_x, r_c) \tag{3.7}$$

Let Θ_P denote the set of all the possible sets of events τ_P that define a different agent population transition from O^c and that obtains a different O^f. The challenge of the agent population transition is to find the specific set of events $\widehat{\tau}_P$ that minimizes the agent population transition impact:

$$OTI(\widehat{\tau}_P) = \underset{\tau_P \in \Theta_P}{\operatorname{argmin}}\, OTI(\tau_P) \tag{3.8}$$

The application of the agent population transition algorithm calculates the set of events of the minimal impact $\widehat{\tau}_P$ and the final organization O^f that is to be transitioned to by applying this set of events. Each execution of the algorithm (line 7 of Algorithm 1) takes into account the addition or deletion of a single agent and relationship but several iterations can be carried out if several executions are carried due to the loop (lines 4-9 of Algorithm 1).

The transition path of the minimal impact for the agent population transition, defines a transition from O^c to O^f, which has associated the set of $agents$, $acquaintance$, $provides$, and $plays$ events of the minimal impact $I(\widehat{\tau}_A)$.

3.5 Conclusions

The Organization Transition Model presented in this chapter provides support for representing how a transition between the current organization and a future instance can be carried out at the minimal impact. This impact provides the required information so that the organization can deliberate on the suitability of carrying out the transition according to the costs associated to the process and the benefits obtained. Furthermore, the specification of costs is carried out for each specific element that can be changed, which provides a more realistic evaluation of the benefits and costs.

In contrast to other previous works, our model takes into consideration several penalizations such as the number of roles that each agent plays in order to compute the cost or the benefits of playing a role. As an example, the work of [75] reallocates each role to the agent that maximizes the utility by playing the role, independently of the number of roles that the agent already plays. We consider that this restriction could not be applied to a realistic scenario in which the utility of an agent for playing a role (and also, the cost associated to it) should be different depending on the number of roles that the agent is currently playing.

Considering multiple transitions for reorganization offers a more wide range of solu-

tions than other works that only considers changes in the roles played by agents, in the relationships between agents, or in the population of agents. The multi-dimensional criteria for deliberation presented in this chapter provide better decisions for reorganization. In the next chapter we show how this model is incorporated into a mechanism that obtains the best decision for reorganization. We provide the algorithms that are required to calculate each organization transition as well as the multi-dimensional deliberation and the steps that are required to carry out the transition.

CHAPTER

4 Multi-dimensional Transition Deliberation Mechanism

4.1 Multi-dimensional Transition Deliberation Mechanism 84
4.2 Role Reallocation Transition 86
4.3 Acquaintance Transition 89
4.4 Agent Population Transition 90
4.5 Deliberation 91
4.6 Calculating the transition path 92
4.7 Conclusions 94

In this chapter, we present the mechanism that allows to adapt the organization in several organization dimensions. This mechanism is called Multi-dimensional Transition Deliberation Mechanism (MTDM) and provides a decision-making support that considers the three types of transitions: role reallocation transition, acquaintance transition, and agent population transition. By specifying the requirements of the final organization that is to be achieved, the MTDM accurately predicts the impact of the transition in terms of two aspects: the costs associated to the organization transition, and the benefits or costs that this transition causes not only to the agents involved in the change but also to the whole organization. The MTDM provides support to

the adaptation of organizations by taking into account the potential for improvement in utility based on the costs associated to the process. Moreover, since several transitions on different dimensions are considered, the range of adaptation solutions is increased.

The MTDM allows a specific organization to evolve into another organization given some requirements, at the minimal OTI. It also provides the sequence of steps that must be carried out to achieve the future organization by taking into account the restrictions that must be fulfilled during the transition. First, we present an overview of the complete process. Then, we describe in detail the algorithms that are used for the three types of transitions, for the deliberation strategy, and for obtaining the transition path of the minimal OTI.

4.1 Multi-dimensional Transition Deliberation Mechanism

The MTDM is a multi-stage mechanism that is based on the models used in the strategic management research area for analyzing the performance of business firms [103]. The model proposed by Zolt is aimed at analyzing the firm's performance by simulating the consequences when several changes in resources, operational routines, or competencies, are carried out. These changes are proposed through imitation or experimentation.

Similarly, the MTDM calculates transitions in different dimensions to other organizations with high expected utility based on the cost for transition to these organizations. The benefits and costs of transition are measured in terms of OTIs. Then, the MTDM decides which transition is finally implemented and provides the sequence of changes required to carry out the transition. Algorithm 1 represents the main cycle of the MTDM, which is composed of different stages that are summarized in the following sections.

Algorithm 1: Transition Deliberation Mechanism

1: INPUT: O^c
2: OUTPUT: τ, $OTI(\tau)$
3: $O^f \leftarrow O^c$
4: **do**:
5: $O_R \leftarrow role_reallocation_transition(O^f)$
6: $O_A \leftarrow acquaintance_transition(O^f)$
7: $O_P \leftarrow agent_population_transition(O^f)$
8: $O^f, OTI \leftarrow deliberation(O_R, O_A, O_P, O^f)$
9: **while** OTI is improved
10: $\tau \leftarrow Transition_path(O^c, O^f)$
11: **return** $\tau, OTI(\tau)$

- The first step calculates transitions in multiple dimensions from the current organization O^c. These transitions correspond to the multi-dimensional transitions considered in the Organization Transition Model: role reallocation transition, acquaintance transition, and agent population transition. Lines 5-7 obtain three different future organizations that could be achieved from O^c with high expected utility based on the cost for transition to these organizations. The benefits and costs of transition are measured in terms of OTIs.

- With the three organizations obtained from the previous step, a deliberation process (line 8) selects which of these three organizations has associated the high expected utility based on the transition costs. This process is carried out iteratively while an organization is found, which can be achieved with a better OTI. Due to a loop execution is introduced, several changes in different dimensions can be selected through different iterations. Since the number of changes are limited and the impact improvement is lower each iteration, a solution is found in a bounded number of iterations. If there is not any future organization that can be achieved with a negative impact, the best decision is not transition (the current organization is maintained).

- After obtaining the future organization O^f, which is to be transitioned to, a sequence of required events τ is obtained (line 10). These events allow to

transform the current organization O^c into O^f. Then, the dependency of events of τ is also calculated. In this case, dependent events must be split into different subsets, providing a sequence that must be applied in order of dependence by defining the transition path between both organizations.

4.2 Role Reallocation Transition

Algorithm 2 calculates the final organization O^f, which can be transitioned from O^c at the minimal OTI by a role reallocation transition. In this implementation, the organizational objects, the relationships $offers$, and $acquaintance$ are the same for both organizations, and changes in the relationships $provides$ and $plays$ represent the role reallocation transition.

To implement this algorithm, we used a modification of the the Hungarian algorithm [68], which solves problems of task assignment in polynomial time. The classic version of the algorithm solves problems given n workers and n tasks. Our modification deals with problems that involve computation of non-square matrixes. The requirement of this mechanism is to find the role reallocation to agents that minimizes the impact, thus other algorithms of task assignment could also be used. This problem is similar to problems of resource allocation studied in other research areas such as the work of Dolgov & Durfee [38].

The algorithm calculates a matrix of impacts $I = |A| \times |R|$ (lines 5-7), which represents the impact of role reallocation of each agent to each role according to the services offered by each role and the services provided by each agent (as explained in Section 3.4.1). $|A|$ is the number of agents in the organization and $|R|$ is the number of different roles of the organization. The element in the ith file and j^{th} column represents the impact of the agent located in row i to play the role located in column j.

The first iteration of the algorithm attempts to reallocate each agent to the role that

Algorithm 2: Role Reallocation Transition

```
1:  INPUT: O^c
2:  OUTPUT: O^f
3:  for each a ∈ A^c do:
4:      r^c ← r ∈ R^c | (a, r) ∈ plays^c
5:      for each r ∈ R^c do:
6:          I_{a,r} ← I_R(a, r^c, r)
7:      end for
8:  end for
9:  for each agent a and role r do:
10:     I_{a,r} ← I_{a,r} − min(a) − min(r)
11: end for
12: while ¬optimal(I) do:
13:     mark_ag = mark_r = ∅
14:     mark_ag ← ¬assigned_ag(I)
15:     while change(mark_ag) ∨ change(mark_r) do:
16:         mark_r ← mark_r ∪ roles(I, 0)
17:         mark_ag ← mark_ag ∪ assigned_ag(mark_r)
18:     end while
19:     min_value ← min(mark_ag, ¬mark_r)
20:     for each I_{a,r} do:
21:         if a ∈ mark_ag ∧ r ∈ mark_r then
22:             I_{a,r} ← I_{a,r} + min_value
23:         end if
24:         if a ∉ mark_ag ∧ r ∉ mark_r then
25:             I_{a,r} ← I_{a,r} − min_value
26:         end if
27:     end for
28: end while
29: plays^f ← reallocation(I)
30: for all (a, r) ∈ plays^f do:
31:     if (a, r) ∉ plays^c do:
32:         for all (r, s) ∈ offers^f do:
33:             provides^f ← add_provides(provides^c(a, s))
34:         end for
35:     end if
36: end for
37: O^f ← ⟨R^c, S^c, A^c, offers^c, provides^f, plays^f, acquaintance^c⟩
38: return O^f
```

requires the lowest transition impact. The impacts for each agent are updated by taking into account these minimal impacts (line 10). If an optimal assignation cannot be made by taking into account the requirements of the final organization (line 12), the algorithm proceeds as follows:

- **Step 1:** Mark the rows of agents that are not assigned to any role because their lowest role reallocation transition impact has been assigned to one or more other agents (line 14).
- **Step 2:** Mark the columns of roles that have the lowest reallocation impact for the previously marked agents (line 16).
- **Step 3:** Mark the rows of agents that are assigned to the previous roles (line 17).
- **Step 4:** Repeat steps 2 and 3 while new rows are marked.
- **Step 5:** Calculate the minimal value of the matrix by taking into account the impacts of the marked rows and unmarked columns (line 19).
- **Step 6:** Recalculate the impacts of the matrix by adding the minimal value to the elements of the marked rows and columns and subtracting the minimal value of the unmarked rows and columns (lines 20-27).

Once the optimal assignment has been obtained, the specification of O^f is completed with the new $plays$ and $provides$ relationships. In order to calculate the optimal assignment (line 12), we use an implementation of the Edmonds-Karp algorithm [39], which has been applied in the literature for these kinds of problems. This algorithm computes the maximum flow in a flow network in $O(|N| \cdot |A|^2)$ (with N being the nodes and A being the arcs of the flow network) by using a breadth-first search.

Finally, the $provides$ relationships of the final organization O^f are calculated depending on the services offered by each role and the role played by each agent (lines 30-36).

4.3 Acquaintance Transition

Algorithm 3 calculates the final organization O^f, which can be transitioned from O^c at the minimal OTI by an acquaintance transition. In this implementation, the organizational objects, the relationships $offers$, $provides$, and $plays$ are the same for both organizations, and changes in the $acquaintance$ relationships represent the acquaintance transition.

To implement this algorithm, we used an iterative version which obtains the most profitable acquaintance addition or deletion given the current state of the organization O^c. The solution obtained do not consider changes that cause transitions to organization which do not satisfy the domain restrictions (e.g. if the maximum number of acquaintances for each agent is achieved).

Algorithm 3: Acquaintance Transition

1: INPUT: O^c
2: OUTPUT: O^f
3: $min_impact \leftarrow 0$
4: **for all** $a \in A^c$ do:
5: **for all** $b \in A^c \wedge b \neq a$ do:
6: **if** $(a, b) \in acquaintance^c$ do:
7: $acquaintance^i \leftarrow del_acq(acquaintances^c, (a, b))$
8: **else**:
9: $acquaintance^i \leftarrow add_acq(acquaintances^c, (a, b))$
10: **end if**
11: $O^i \leftarrow \langle R^c, S^c, A^c, offers^c, provides^c, plays^c, acquaintance^i \rangle$
12: $\tau \leftarrow calc_events(O^c, O^i)$
13: **if** $OTI(\tau) < min_impact$ then:
14: $min_impact \leftarrow OTI(\tau)$
15: $O^f \leftarrow O^i$
16: **end if**
17: **end for**
18: **end for**
19: **return** O^f

This algorithm explores the different acquaintances modifications depending on the

structural topology of the current organization O^c. The addition and deletion of a single acquaintance between a pair of agents (lines 7 and 9, respectivelly) cause an acquaintance transition to a final organization O^i. For each final organization that can be achieved, the addition and deletion events required to transform O^c into O^i are obtained (line 12). Finally, this algorithm returns the final organization O^f that can be achieved by an acquaintance modification that has associated the minimal impact. It is assumed that the addition and deletion functions return a set of valid acquaintances that fulfills any organizational domain-restriction (e.g. these functions are not applied if the number of acquaintances per agent is violated).

4.4 Agent Population Transition

Algorithm 4 calculates the final organization O^f, which can be transitioned from O^c at the minimal OTI by an agent population transition. In this implementation, the organizational objects $roles$ and $services$, and the organizational relationships $offers$ are the same for both organizations, and changes in the rest of organizational elements are considered.

This algorithm explores the range of final organizations that can be achieved when any agent is deleted from the organization (line 5). The deletion of an agent modifies the A objects, and may also modify other elements such as $plays$, $provides$, and $acquaintance$, depending on the roles, services, and relationships that are associated to this agent. Similarly to the acquaintance transition, the deletion function is applied when the domain restrictions are satisfied. The algorithm also considers the addition of a new agent b into the organization (line 13). In this case, it is assumed that agent b is known by the organization and the addition impact can be measured. Depending on the domain restrictions, the addition of a new agent may also cause that one ore more services are assigned to be provided by the agent, some role is assigned to be played by the agent, and acquaintances are assigned. However, the solution obtained do not consider several changes if these are not required by the domain restrictions (e.g. if

the minimum number of acquaintances for each agent is 1, the algorithm relates the new agent only with one existing agent). The algorithm returns the final organization O^f that has associated the minimal OTI.

Algorithm 4: Agent Population Transition

INPUT: O^c, b
OUTPUT: O^f
$min_impact \leftarrow 0$
for all $a \in A^c$ do:
    $A^i, provides^i, plays^i, acquaintance^i \leftarrow del_agent(O^c, a)$
    $O^i \leftarrow \langle R^c, S^c, A^i, offers^c, provides^i, plays^i, acquaintance^i \rangle$
    $\tau \leftarrow calc_events(O^c, O^i)$
    if $OTI(\tau) < min_impact$ then:
        $min_impact \leftarrow OTI(\tau)$
        $O^f \leftarrow O^i$
    end if
end for
$A^i, provides^i, plays^i, acquaintance^i \leftarrow add_agent(O^c, b)$
$O^i \leftarrow \langle R^c, S^c, A^i, offers^c, provides^i, plays^i, acquaintance^i \rangle$
$\tau \leftarrow calc_events(O^c, O^i)$
if $OTI(\tau) < min_impact$ then:
    $min_impact \leftarrow OTI(\tau)$
    $O^f \leftarrow O^i$
end if
return O^f

4.5 Deliberation

Once the organizations that minimizes the OTI for each dimension are calculated, the next stage of the MTDM (line 8 of Algorithm 1) decides which transition is finally implemented depending on the deliberation strategy (Algorithm 5). The current implementation is focused on selecting the transition that minimizes the OTI by considering several changes (line 7).

Algorithm 5: Deliberation

1: INPUT: O^R, O^A, O^P, O^c
2: OUTPUT: O^f
3: $min_impact \leftarrow 0$
4: $\tau^R \leftarrow calc_events(O^c, O^R)$
5: $\tau^A \leftarrow calc_events(O^c, O^A)$
6: $\tau^P \leftarrow calc_events(O^c, O^P)$
7: $min_impact \leftarrow min(OTI(\tau^R), OTI(\tau^A), OTI(\tau^P))$
8: $O^f \leftarrow min(O^R, O^A, O^P)$
9: **if** $min_impact < 0$ then:
10: **return** O^f, min_impact
11: **end if**
12: **return** $O^c, 0$

Even though a single change is carried out by an individual execution of the acquaintance transition algorithm and the agent population transition algorithm, several changes can be obtained by executing different iterations of the algorithms (lines 4-9 of Algorithm 1). Thus, the future organization that is selected to be transitioned once the loop ends, can be composed of combination of transitions (e.g. changing relationships between some pairs of agents and then, swapping the roles played by some of these agents). Since the number of changes are limited and the impact improvement is lower each iteration, a solution is found in a bounded number of iterations. The organization that is obtained after deliberation, represents the future organization O^f with the high expected utility based on the transition costs and considers changes in different dimensions. If there is not any future organization that can be achieved with a negative impact, the best decision is not transition.

4.6 Calculating the transition path

Once the final organization O^f that is transitioned to is selected, the final stage (Algorithm 6) obtains the specific sequence of events τ that allow this transition from O^c to O^f and the impact associated to applying these events $OTI(\tau)$ (line 10 of Algorithm

1).

Algorithm 6: Transition path

```
1:  INPUT: O^c, O^f
2:  OUTPUT: τ (ordered)
3:  τ ← calc_events(O^c, O^f)
4:  subset = 1
5:  τ_subset ← τ
6:  while τ_subset ≠ ∅ do:
7:      for all ε_i ∈ τ_subset do:
8:          D ← calculate_dependencies(ε_i, τ_subset)
9:          if D ≠ ∅ then
10:             move(ε_i, τ_subset+1)
11:         end if
12:     end for
13:     subset++;
14: end while
15: return τ
```

First, the addition and deletion events required to transform O^c into O^f are calculated for each organizational component (line 3). These events represent objects that are added and deleted from O^c_O in order to obtain O^f_O, and relationships that are added and deleted from O^c_R in order to obtain O^f_R.

Finally, the last step is to find the transition path between O^c and O^f according to the dependency among the events of τ. The algorithm splits two dependent events until a sequence of subsets of independent events is found. The application of the sequence of subsets defines the transition path. This algorithm returns the ordered sequence of subsets of events $\tau_i \in \tau$ that allows a transition between O^c and O^f by transitions to intermediate organizations if they are required.

4.7 Conclusions

The MTDM presented in this chapter provides a deliberation mechanism for organization adaptation based on a multi-dimensional transition criteria. The organization transition considers the impact of transition in terms of the utility caused by the transition, the costs associated to the transition, and how this transition would influence all the components of the organization.

The MTDM provides an accurate estimation of the transition impact since the organization that is to be achieved is calculated by each transition. Thus, the impact associated to each change that is required to carry out the transition, can be measured individually and more accurately than other approaches. The suitability of the adaptation must be considered taking into account not only the benefits obtained by adaptation but also the costs associated to this process. This issue is also important in human organizations since most organizational changes may encounter problems: they often take longer than expected and desired; the cost of managerial time may be increased; and there may be resistance from the people involved in the change [67]. In the next chapter, we present the implementation of the adaptation framework that provides this MTDM.

CHAPTER

5 Reorganization Facilitator Service

5.1 Magentix Multiagent Platform 95
5.2 Reorganization Facilitator Service 97
5.3 Conclusions 101

In this chapter we present the framework that implements the MTDM: the Reorganization Facilitator (RF) Service. This service has been implemented in the Magentix Multiagent Platform [41] and provides the support for agent interaction in order to obtain which organization transitions can be carried out at the minimal OTI and how this process must be applied. First, we present an overview of the Magentix Multiagent Platform, and then, the description of the RF.

5.1 Magentix Multiagent Platform

We use Magentix as the framework for developing the *RF* service. Magentix supports and enables the development and execution of open multiagent systems and it is developed in Java. It focuses on providing support at the interaction and organization levels, which are key levels in open environments, where heterogeneous agents

interact and organize themselves into organizations.

The support that Magentix provides for agent interactions is composed of two main parts: agent communication and agent conversations. Magentix uses the Advanced Message Queue Protocol (AMQP) standard [1] as a foundation for its communication infrastructure. It allows heterogeneous agents to interact with each other via messages that are exchanged using this standard. Interactions among agents in Magentix are oriented to conversations. A conversation between agents is represented by a sequence of messages following a specific Interaction Protocols. Magentix facilitates the specification, automatic execution and management of Interaction Protocols which an agent is carrying out. These protocols can be dynamically modifyed according to the requirements of the system. On the one hand, agent organizations have associated norms according to what a member is permitted to do, prohibited from doing and is obligatory according to her role and relationships with other members. On the other hand, interactions between agents are carried out by IP specifications which represent the communication rules.

Magentix provides support to virtual organizations by means of the THOMAS architecture [84], which defines flexible services that can be used by agents:

- Service Facilitator (*SF*), which allows the registration and search of services provided by internal or external entities by following *Service-Oriented Architectures* guidelines.

- Organization Management System (*OMS*), which is in charge of the management of the organizations, taking control of their underlying structure, the roles played by the agents, and their relationships.

The *SF* and the *OMS* provide services for managing the life-cycle of the organizations as well as the services provided by the agents. Therefore, systems can be developed where agents are able to dynamically enter and leave the system, change their services, their relationships, or the roles that they play in the organization.

Magentix also incorporates modules for providing a tracing service and security support. On the one hand, the Tracing module allows agents to share information in an indirect way by means of trace events. On the other hand, the Security module provides support for building a secure MAS by means of features such as privacy, opennes and interoperability.

5.2 Reorganization Facilitator Service

The RF service has been implemented as a new service of Magentix (Figure 5.1). This service provides the support for the different phases of the MTDM.

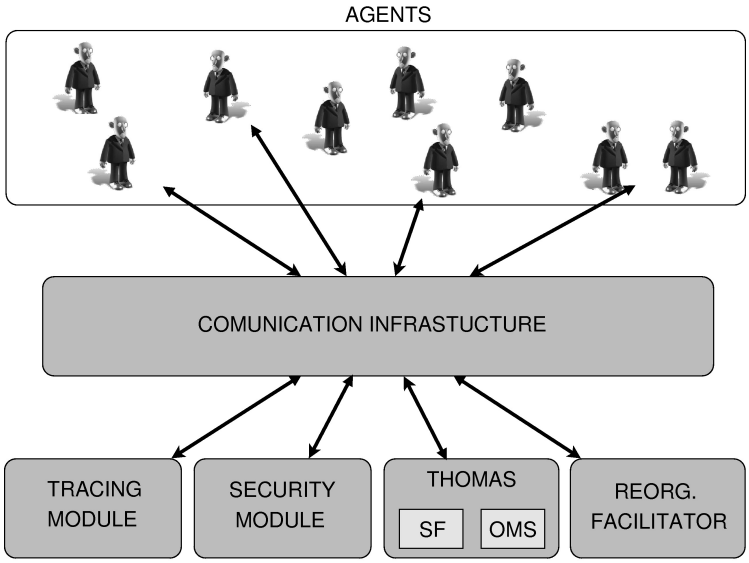

Figure 5.1: Magentix Architecture

Each organization in Magentix has associated an specific agent which acts as the *manager* of the organization. This agent is the responsible of interacting with the *SF* and the *OMS* services for managing the life-cycle of the organization. This agent will also the responsible of interacting with the *RF* service in order to obtain the sequence

of steps required to transition from the current organization to a future one with the minimal OTI.

In Figure 5.2 we show the internal modules of the RF, which implement the algorithms described in previous Chapter 4.

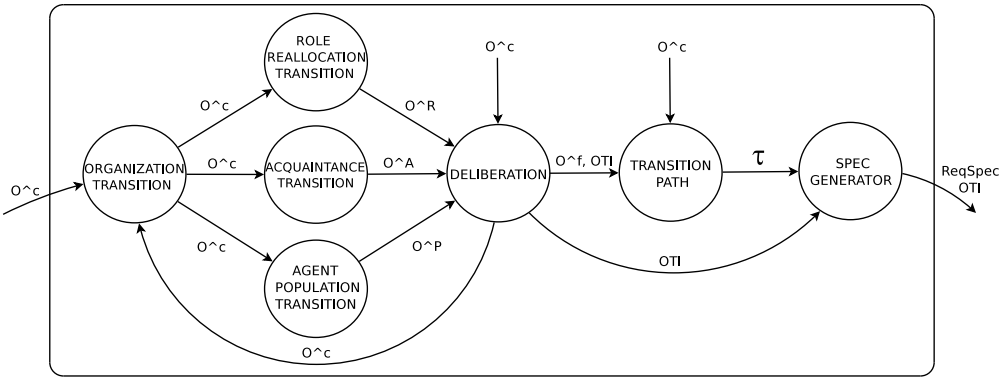

Figure 5.2: Reorganization Facilitator Service

Given the current organization O^c, the three modules related to the transitions obtain three different organizations that can be transitioned to, at the minimal OTI (O^R, O^A, and O^P), then, the Deliberation module is able to combine changes in different dimensions while the OTI can be decreased. When the final organization O^f is obtained, the sequences of steps are calculated by the Transition path module. Finally, the Specification generator module translates these sequence of steps in a xml specification that can be used by the organization manager in order to carry out the transition.

Being in charge of coordinating every reorganization process in an agent organization, the manager agent estimates the impact for each potential change. This impact represents the costs/benefits that the application of an individual change (such as the addition or deletion of a service) would cause, not only to those components involved in the change, but also to other components in the organization. Furthermore, it also shows the cost for carrying out the application of this change.

Individual impacts which are calculated by the organization manager agent are transferred to the *RF* in order to calculate the organization transition. The *RF* manages the impacts defined to the different events related to an organization transition.

This makes the *RF* interact with the *OMS* to retrieve information regarding the organization that is to be to transitioned from. The *RF* finds the organization with the minimal OTI and determines the sequence of steps required to achieve it. Then, the agent can ask the *OMS* and the *SF* services to carry out this organization transition (Figure 5.3).

```
register_transition_impacts(OrgID ?OID, CostSpec ?spec)
```

The *?OID* parameter represents the identifier of the current organization, and the *?spec* parameter represents the specification of the impacts.

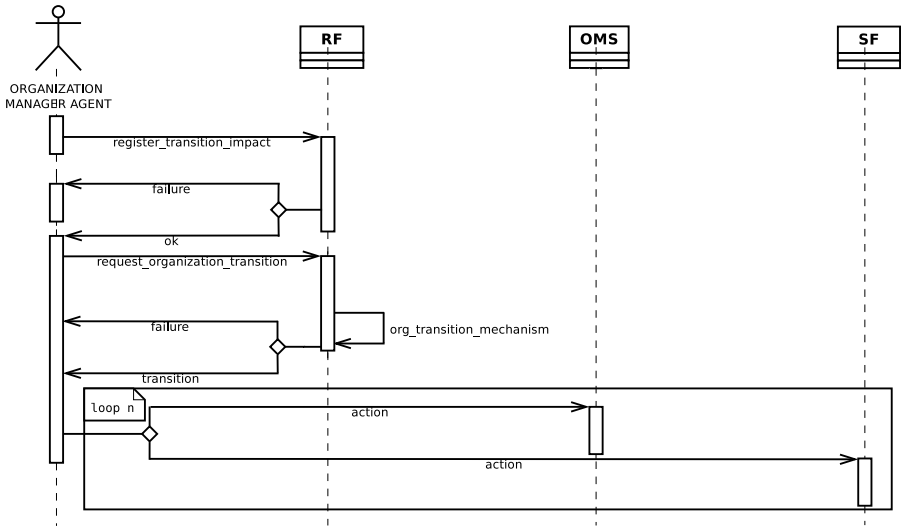

Figure 5.3: Reorganization Facilitator Interaction

The *RF* also provides services for assessing the impact of an individual objects and relationships:

```
register_add_role_impact(OrgID ?OID, RoleID ?RID, Impact ?impact)
```

```
register_delete_role_impact(OrgID ?OID, RoleID ?RID, Impact ?impact)
register_add_service_impact(OrgID ?OID, ServID ?SID, Impact ?impact)
register_delete_service_impact(OrgID ?OID, ServID ?SID, Impact ?impact)
register_add_agent_impact(OrgID ?OID, AgentID ?AID, Impact ?impact)
register_delete_agent_impact(OrgID ?OID, AgentID ?AID, Impact ?impact)
register_add_offers_impact(OrgID ?OID, RoleID ?RID, ServID ?SID,
                           Impact ?impact)
register_delete_offers_impact(OrgID ?OID, RoleID ?RID, ServID ?SID,
                           Impact ?impact)
register_add_provides_impact(OrgID ?OID, AgentID ?AID, ServID ?SID,
                           Impact ?impact)
register_delete_provides_impact(OrgID ?OID, AgentID ?AID, ServID ?SID,
                           Impact ?impact)
register_add_plays_impact(OrgID ?OID, AgentID ?AID, RoleID ?RID,
                           Impact ?impact)
register_delete_plays_impact(OrgID ?OID, AgentID ?AID, RoleID ?RID,
                           Impact ?impact)
register_add_acquaintance_impact(OrgID ?OID, AgentID ?AID, AgentID ?AID,
                           Impact ?impact)
register_delete_acquaintance_impact(OrgID ?OID, AgentID ?AID,
                           AgentID ?AID, Impact ?impact)
```

Depending on the specific service, the *?SID*, *?RID*, and *?AID* parameters are the service, role, and agent, respectively, for the specific event. The *?impact* parameter defines the impact of this event. As an example, we can define i as the impact of agent a providing the service s in the organization o, using the following request to the RF:

```
register_add_provides_impact(o,a,s,i)
```

We must point that some of the above requests are allowed to receive special parameters such as *?ANY*, specially useful for the addition of objects, in order to represent that every event of the specific type has associated the same impact:

```
register_add_role_impact(o,ANY,i)
```

The above request would represent that the addition event of a role in the organization o has associated an impact i.

Once these events are specified, the *RF* can be asked to calculate an organization transition using the following service:

```
request_organization_transition(OrgID ?OID)
```

The *?OID* parameter defines the identifier of the organization to be transitioned from. The tool EMFGormas [43] provides support for specifying this organization.

The *RF* requests the *OMS* for the information regarding the current organization. Then, the *RF* is able to calculate the sequence of events that causes a transition to the future organization with the lowest OTI. Finally, these events are returned to the organization manager agent in a xml specification, which is used to ask the OMS and the SF services to carry out this organization transition.

5.3 Conclusions

The RF service presented in this paper supports the MTDM. This allows organizations to calculate transitions according to the transition impacts that are defined. The agent manager, which is the responsible of the organization management in Magentix, is also used as the responsible for interacting to the RF service. In the next chapters we introduce two case studies in order to test and validate the model, the mechanism, and the framework proposed. These tests are divided in two real examples which represent an application of touristic services (Chapter 6) and an application on a service providers network (Chapter 7).

Part IV

Case Studies

CHAPTER

6 Organization Transitions in a Tourist Application

6.1 Introduction 106
6.2 Impact Measurement of Organization Transition 108
6.3 Trace 111
6.4 Performance Evaluation 113
6.5 Execution 120
6.6 Conclusions 126

In order to evaluate the scalability, the efficiency, and the performance of the RF, in this chapter we present a first example which consists on an application of touristic services. After describing the test-bed scenario, we focus the experiments on a specific type of transition: the role reallocation transitions. First, we present a trace of execution in order to show the interaction between the organization manager and the RF. Second, we test the scalability and the efficiency of the underlying algorithms that are implemented in the RF service. Finally, we test the effectiveness of the role reallocation transition by comparing the organization performance under different conditions of adaptation.

6.1 Introduction

The tourist application is composed of agents that are grouped into three different organizations: user agents, broker agents, and provider agents. User agents require tourist services and request information regarding different travel packages, which involves booking of hotels, flights, trains, etc. These agents interact with broker agents in order to obtain the reservations required. Provider agents are the agents that belong to the specific hotels, airlines, and train companies, etc. Broker agents are in charge of offering travel packages that are based on their negotiation with different provider agents. Broker agents act as intermediaries between user agents and provider agents. In this example, we focus on the organization of broker agents.

The organization of broker agents at a moment t is defined as $O^t = \langle O^t_O, O^t_R \rangle$. The Organizational Objects $O^t_O = \langle R^t, S^t, A^t \rangle$ defines the individual objects of the organization. In this application, six different roles are defined as $R^t = \langle r_0, r_1, r_2, r_3, r_4, r_5 \rangle$ for every moment t. Each one of these roles is specialized for a specific kind of travel package: honeymoon & romance (r_1), spa & wellness (r_2), business (r_3), sports activities (r_4), cultural & sightseeing (r_5), and family (r_6). Each role offers a service that is related to the management of the specific kind of travel packages that the role is specialized in. As an example, role r_1 offers the service s_1 which is the honeymoon & romance travel package management service.

The population of broker agents is defined as $A^t = \{a_1, \ldots, a_n\}$ for a given moment t. At a moment t, an agent a_x plays the role r_y if the service s_y is provided by a_x at that moment. Nevertheless, this agent can provide other services and can play different roles that it is not currently playing at a given cost. This cost represents the penalization for returning the travel packages that have not been sold.

In a period of time between t and t', each broker agent a_x, which plays the role r_y, receives a number of requests from user agents that demand travel packages related to the service s_y. This number of requests is represented as $R(a_x, s_y)^{t'}_t$. Each broker agent a_x has an associated suitability for providing a service s_y which is defined as

$suitability(a_x, s_y)$. This suitability defines how well a broker agent can provide a service and, thus, how well this agent can play a role.

From all the requests received at an agent a_x for the service s_y, a specific number of these requests will finally be sold, which is represented as $S(a_x, s_y)_t^{t'}$, and corresponds to the profit that a_x generates to the organization. This number depends on the suitability of a_x for providing s_y, which in this example, is represented as the probability of a received request finally being sold. Apart from the profit, suitability also influences the costs associated for leaving a role.

The aggregation of the requests received by every agent that plays the role r_y represents the number of requests received in the whole organization:

$$R(A, s_y)_t^{t'} = \sum_{a_x \in A^t} R(a_x, s_y)_t^{t'} \mid plays(a_x, r_y) \in plays^t$$

In order to simulate this application, each request has associated the same probability $p = \frac{1}{|plays_{r_y}^t|}$ of being received by each agent a_x that plays the role r_y, where $|plays_{r_y}^t|$ is the total number of agents that play r_y at t.

The number of sales are represented as a random variable X that follows a binomial distribution $X \sim B(n, p)$, with parameters $n = R(a_x, s_y)_t^{t'}$ and $p = suitability(a_x, s_y)$. In order to simulate the number of sales carried out by each agent from the number of requests received, a Bernoulli random numbers generator (Algorithm 10) is used. This method is similar to the one used in [58] and generates n independent random values, returning the number of these values that are less than or equal to p.

The utility of each agent a_x that plays the role r_y during the period of time between t and t' is represented as the number of sales by this agent multiplied by the profit obtained from the sale of each individual travel package of the service s_y:

$$U(a_x)_t^{t'} = S(a_x, s_y)_t^{t'} \times profit(s_y)$$

The utility of the organization between two time-steps t and t' is measured as the

Algorithm 10: Bernoulli random numbers generator

```
1:  INPUT: n, p
2:  OUTPUT: x
3:  x ← 0,  k ← 0
4:  while k ≤ n do:
5:      Generate r ∼ random(0, 1)
6:      k ← k + 1
7:      if u ≤ p then x ← x + 1
8:      end if
9:  end while
10  return x
```

profit generated by all the sales of the organization in this period:

$$U(O)_t^{t'} = \sum_{a \in A} U(a_x)_t^{t'}$$

The objective of the organization is to maximize this utility. As we stated at the beginning of this section, in this example we focus on role reallocation transitions. At design time, it may not be possible to know what service distribution among the agents provides the highest utility. Furthermore, even though the best performing role allocation is known, it may not be the best one at every moment of the organization's life-span because service demand may change, requiring a different distribution. Therefore, organization transitions provide alternatives for improving utility. In the following section, we illustrate how role reallocation costs and benefits are measured in terms of impacts.

6.2 Impact Measurement of Organization Transition

The impact of role reallocation for agent a_x is obtained by Eq. 3.3. This equation calculates the impact of leaving the current role (Eq. 3.2) and playing a new role (Eq. 3.1). The term $I_A(a_x, r_n)$ (Eq. 3.1) represents the impact of agent a_x providing the

services offered by the new role r_n. In this example, we consider that each broker agent is able to provide each service at a given suitability. Thus, this impact can be assumed to be null since no cost is required to provide a service.

The term $i(add_plays((a_x, r_n)))$ (Eq. 3.1) measures how the organization utility would be affected if a_x plays the role r_n. According to the law of large numbers [14], if a large number of requests are received, these would be uniformly distributed among all the agents that play the role r_y. Being $R(A, s_n)_t^{t'}$ the requests received for the service s_n in the organization in a previous period of time, the number of these requests that would be received by a_x can be estimated as the number of requests divided by the number of agents that would be playing r_n: $\frac{R(A,s_n)_t^{t'}}{|plays_{r_n}^{t'}|+1}$.

Given these received requests and the suitability of a_x for providing the service s_n, the number of sales that would be carried out by a_x is estimated as $S(a_x, s_n)_t^{t'}$ by using Algorithm 10 (Section 6.1).

In contrast, these requests would not be received by the agents that were already playing r_n, causing these requests to be reduced to:

$$R(A, s_n)_t^{t'} - \frac{R(A, s_n)_t^{t'}}{|plays_{r_n}^{t'}| + 1} = \frac{R(A, s_n)_t^{t'} \times |plays_{r_n}^{t'}|}{|plays_{r_n}^{t'}| + 1}$$

Therefore, the benefits generated by the agents that were already playing r_n would be reduced to:

$$\frac{S(A, s_n)_t^{t'} \times |plays_{r_n}^{t'}|}{|plays_{r_n}^{t'}| + 1}$$

According to this, the estimated number of sales by all the agents that would be playing r_n can be calculated as:

$$S'(A, s_n)_t^{t'} = S(a_x, s_n)_t^{t'} + \frac{S(A, s_n)_t^{t'} \times |plays_{r_n}^{t'}|}{|plays_{r_n}^{t'}| + 1}$$

According to the above estimations, the impact $i(add_plays((a_x, r_n)))$ is calculated

as the difference between the profits obtained by the sales that are carried out when a_x is not playing r_n and the sales that would be carried out if a_x was playing r_n:

$$(S(A, s_n)_t^{t'} - S'(A, s_n)_t^{t'}) \times profit(s_n)$$

A high impact represents that the utility is expected to decrease if a_x plays the role r_n while a low impact represents an expected increase in utility.

The impact $I_D(a_x, r_c)$ (Eq. 3.2) refers to the cost required for a_x to return the travel packages regarding the service s_c that remain unsold. In this example, agents that provide a service with higher suitability are able to return the unsold travel packages at a lower cost, while agents that provide a service with lower suitability have a higher cost for returning the unsold travel packages. Therefore, the impact $I_D(a_x, r_c)$ is calculated as the number of unsold travel packages that are returned at the price that they should have been sold, according to the agent suitability:

$$unsold(a_x, r_c) \times price(r_c) \times (1 - suitability(a_x, r_c))$$

Finally, the term $i(delete_plays((a_x, r_c)))$ (Eq. 3.2) measures how the organization utility would be affected if a_x is not playing the role r_c. With $R(a_x, s_c)_t^{t'}$ being the number of requests received by a_x for the service s_c, these requests would be received by the other agents that provide s_c if a_x were not playing r_c. Depending on the suitability of these agents, a number of sales would be carried out. The suitability at which these sales would be carried out can be estimated as the mean suitability of the broker agents that were already playing the role r_c from t to t':

$$suitability'(A, s_n) = \frac{S(A, s_c)_t^{t'} - S(a_x, s_c)_t^{t'}}{R(A, s_c)_t^{t'} - R(a_x, s_c)_t^{t'}}$$

Given this suitability and the number of requests $R(a_x, s_c)_t^{t'}$, the sales that would be carried out are estimated as $S'(a_x, s_c)_t^{t'}$ by using Algorithm 10 (Section 6.1). There-

fore, the impact $i(delete_plays((a_x, r_c)))$ is calculated as the difference between the profits obtained by the sales that are carried out if a_x is playing r_c and the profits that would be obtained out if a_x were not playing r_c:

$$(S(a_x, s_c)_t^{t'} - S'(a_x, s_c)_t^{t'}) \times price(s_c)$$

A high impact represents that the utility is expected to decrease if a_x is not playing the role r_c while a low impact represents an expected increase in utility.

6.3 Trace

In this section, we present an example of a transition execution in order to show the interaction between the organization manager and the *RF* service. The objective of this trace is to show the required steps to carry out a transition.

The organization manager agent retrieves information about the requests, the number of sales, and the number of travel packages that remain unsold by each broker agent during a specific period of time. Given the estimation of the impacts described in Section 6.2, the role reallocation impact for each agent a_x from its current role r_c to any other role r_n is obtained according to Eq. 3.3.

By using the service *register_transition_impact* (Section 5.2), the organization manager agent provides the role reallocation impacts to the *RF* service. Then, by using the service *request_organization_transition*, the organization manager agent requests the *RF* service for an organization transition calculation, which obtains the role reallocation transition with the minimal transition impact. If there is not any role reallocation that improves the current configuration, any change is required. Otherwise, the sequence of events that allow the transition to the future organization would be obtained by the *Set_of_events* and *Transition_path* modules. An example is shown below:

```
DeregisterAgentRole(agent1,role4)
```

```
DeregisterAgentRole(agent9,role2)
DeregisterAgentRole(agent10,role0)
RemoveProvider(agent1,service4)
RegisterProvider(agent1,service1)
RegisterAgentRole(agent1,role1)
RemoveProvider(agent9,service2)
RegisterProvider(agent9,service0)
RegisterAgentRole(agent9,role0)
RemoveProvider(agent10,service0)
RegisterProvider(agent10,service4)
RegisterAgentRole(agent10,role4)
```

In this example, the role reallocation consists of three agents swapping roles. Then, the *Spec_generator* module translates this sequence of events into a specification of requests to the *SF* and *OMS* services. A fragment of this specification is shown as follows:

```
1
2  <?xml version="1.0" encoding="UTF-8" ?>
3  <Process>
4    ....
5    <Service_request>
6      <Platform_Service_ID>OMS</Platform_Service_ID>
7        <Service_definition>
8          <Service_name>DeregisterAgentRole</Service_name>
9          <Agent_ID>agent10</Agent_ID>
10         <Role_ID>role0</Role_ID>
11       </Service_definition>
12   </Service_request>
13   ....
14   <Service_request>
15      <Platform_Service_ID>SF</Platform_Service_ID>
16        <Service_definition>
17         <Service_name>RemoveProvider</Service_name>
```

```
18            <Agent_ID>agent4</Agent_ID>
19            <Service_ID>service4</Service_ID>
20         </Service_definition>
21      </Service_request>
22      ....
23   </Process>
```

In case that the organization manager agent decides to finally carry out the transition, the above requests are sequentially carried out in order transition to the future organization in which these three agents swap their roles.

6.4 Performance Evaluation

In this section, we show an evaluation of the *RF* service that is focused on testing the performance and the scalability of the system presented. This evaluation denotes three aspects: (1) how long it takes to find the optimal role reallocation, (2) how the final organization can be achieved, and (3) how often the organization should deliberate on the possibility of transitioning to other organization.

In order to carry out these experiments, some of the constraints defined in Section 6.1 are relaxed. These constraints are related to the number of agents, services, roles, and dependency between events. In this evaluation, the population of broker agents is from 20 to 80 agents ($|A^c| = \{20, 40, 60, 80\}$) for every moment c, with the number of roles and services being half the number of agents ($|R^c| = |S^c| = \frac{|A^c|}{2}$). This is used to test the performance of the system according to the size of the organizational objects. The percentage of services offered by each role from the total number of services provided by the organization changes from 0.2 to 1. Thus, for each role r_y, the percentage of services offered by this role is $p \times |S^c|$, where $p = \{0.2, 0.4, 0.6, 0.8, 1\}$. A percentage of $p = 0.2$ means that each role offers 20% of the services from the whole set of services of the organization. Thus, in order for an agent to be able to play the role, this agent must provide all of these services. This

modification is used to test the performance of the system when the requirements for playing a role are changed.

The dependency between events is another parameter that is changed. A minimal degree of dependency is required to maintain the consistency during the transition process. This dependency is represented as 0 and corresponds to events such as the application of the event $add_plays((a_x, r_n))$ after applying the event $add_provides((a_x, s_n))$. A degree of dependency greater than 0 means additional dependencies that would represent other restrictions such as *"in order to an agent being provider of service s_y, the service s_x must be provided before"*. Therefore, the addition and deletion of events related to these operations are also dependent.

Finally, the costs for providing services and playing roles are also changed in these experiments. Each agent is able to provide only a percentage of the services at a finite cost. Thus, if the impact on the organization of an agent a_x providing a service s_y is infinite ($i(add_provides(a_x, s_y)) = \infty$), this means that the agent is not able to provide this service, and, therefore, it is not able to play the roles that offer this service. This is used to test the performance of the system based on the number of agents that are capable of playing a role. The percentage of services that each agent is able to provide at a finite cost is 20% higher than the number of services offered by each role. With p being the percentage of services offered by a role r_y, each agent is able to provide a percentage of $0.2 + p$ services when $p < 1$, and all the services when $p = 1$.

For each configuration, 1000 different transitions are calculated. Figures 6.1, 6.2, and 6.4 show for each data point the average value of the 1000 organization transitions with a 95% confidence interval. The Student's *t-test* has also been performed for assessing whether the differences based on the parameters are significant or not[1].

[1]We used the R software (`http://www.r-project.org`) to perform the Student's *t-test*.

6.4.1 Results

The purpose of the first experiment is to evaluate the *RF* ability to find the optimal future organization to be transitioned to based on the current organization. Figure 6.1 shows the performance of different agent populations. This performance is shown on the y-axis in terms of iterations of the algorithm used in the *Role_realloc_calculation* module in order to find the optimal role reallocation. The percentage of tourist services offered by each role is shown on the x-axis.

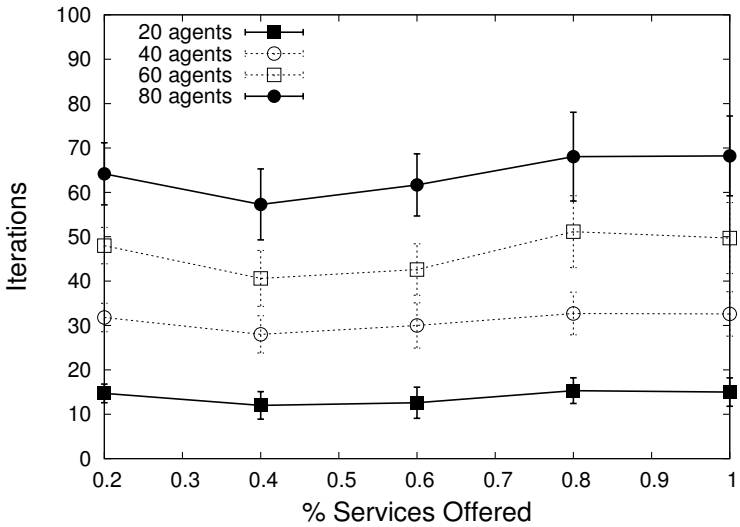

Figure 6.1: Iterations for different agent populations

It can be observed that the agent population that makes up the organization has an influence on the number of iterations carried out by the algortithm. This is because, as the population increases, the number of roles also increases and, in turn, the number of combinations for aquiring roles also increases. Therefore, the algorithm needs to carry out more iterations to achieve the optimal role reallocation. According to the Student's *t-test*, these results are significant.

Figure 6.2 shows a similar experiment but with the agent population on the x-axis. In

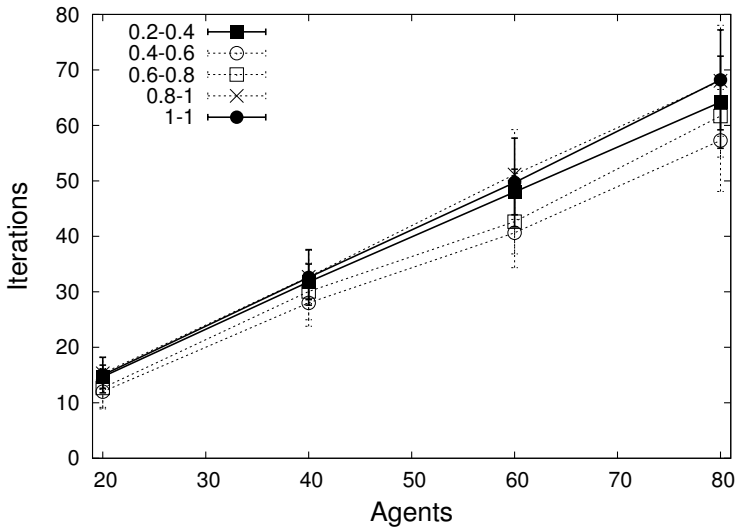

Figure 6.2: Iterations for different percentages of services offered

contrast to Figure 6.1, it can be observed that the number of iterations for the same agent population is similar for each percentage of services offered by the roles (and provided by the agents). According to the Student's *t-test* performed in this experiment, differences are not significant. Therefore, we can conclude that the number of services provided by agents has no influence on the number of iterations that the algorithm carries out for the same agent population.

Figure 6.3 shows the time required for obtaining the organization transition depending on the agent population, the percentage of services offered by a role and the services provided by each agent. It can be observed the increase in the time computation according to the agent population increases. However, it can be pointed out the less computation time when few services are provided by each agent, specially for the largest tested population. This reveals that as the less agents are candidates to play each roles, the less time is required to find best reallocation. In contrast, when each agent is able to play every role, the computation time increases due to all the agents are candidates to be reallocated to each role.

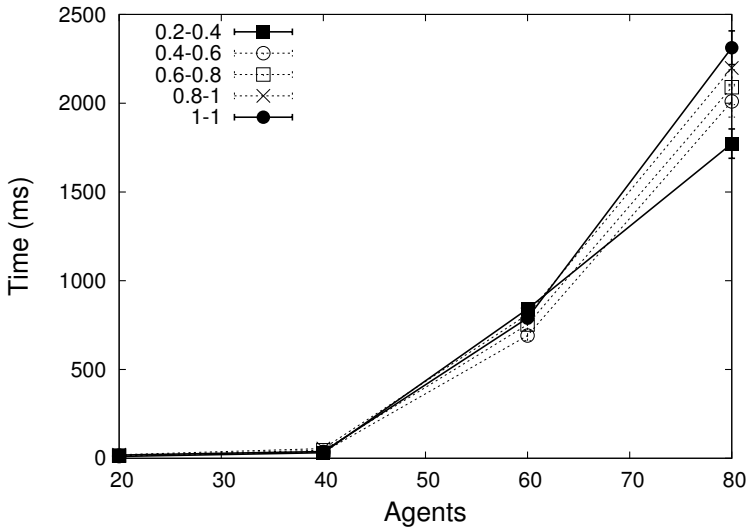

Figure 6.3: Time required to compute transitions

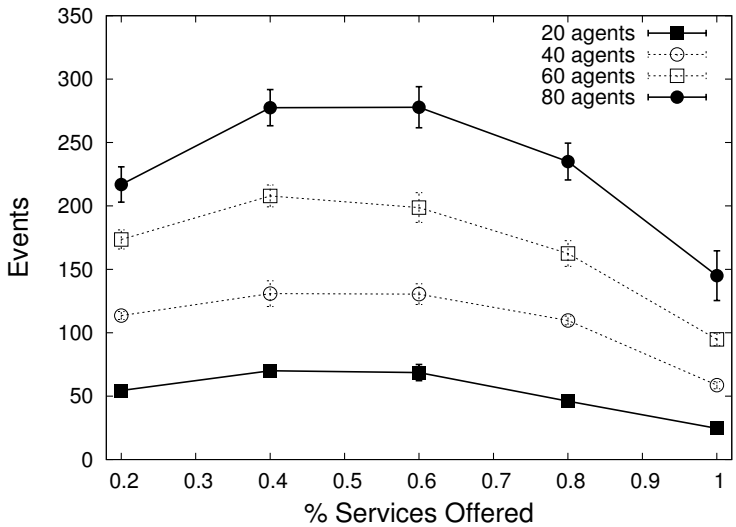

Figure 6.4: Events required for different agent population

Figure 6.4 tests the number of events that are required for transitioning to the future organization. This figure shows the performance of the *RF* service when the per-

centages of services provided by agents and offered by roles are changed. It can be observed that the evolution of the number of events is similar for each agent population. First, the number of events increases until the percentage of services offered by roles is 0.4 or 0.6, and then the number of events decreases as the percentage increases. This behaviour allows us to conclude that, for the same agent population, since broker agents provide a greater number of tourist services, they can play a greater number of roles. Therefore, by reallocating a small number of agents, the transition can be made with a low number of events. When the percentage of tourist services offered by each role is less than 0.6, it means that agents are not able to provide a high number of tourist services. Therefore, if an agent needs to be reallocated, not all the remaining agents are able to play the role left by the agent, and a higher number of reallocations needs to be carried out. In contrast, with a percentage of 0.6 and higher, the agents are quite capable of playing almost every role left by other agents. According to the Student's *t-test*, the size of the broker agent population is significant for the number of events generated.

Figure 6.5 tests the number of intermediate organizations (i.e. the transition path length) required for transitioning depending on the degree of dependency between events. In this experiment the degree of dependency between events is modified in order to test how this parameter influence the transition process. This figure shows on the y-axis the number of intermediate organizations of the algorithm used in the *Transition_path* module for different agent populations. We fix the percentage of tourist services offered by each role to 0.6 and the percentage of services provided by each agent to 0.8. It can be observed that the transition path length increases abruptly when the degree of dependencies changes from 0 to 0.05. However, as this degree increases, the difference in the transition path length does not increases as so much. This reveals that for a specific configuration, we can delimit a maximum number of intermediate organizations that are required to be reached in a transition process. Obviously, as the dependencies between events increase, these must be split into a higher number of subset of events, causing a larger transition path. The differences between a population of 20 agents and the other populations of agents are significant

according the *t-test*, but they are not significant according we increase the number of agents.

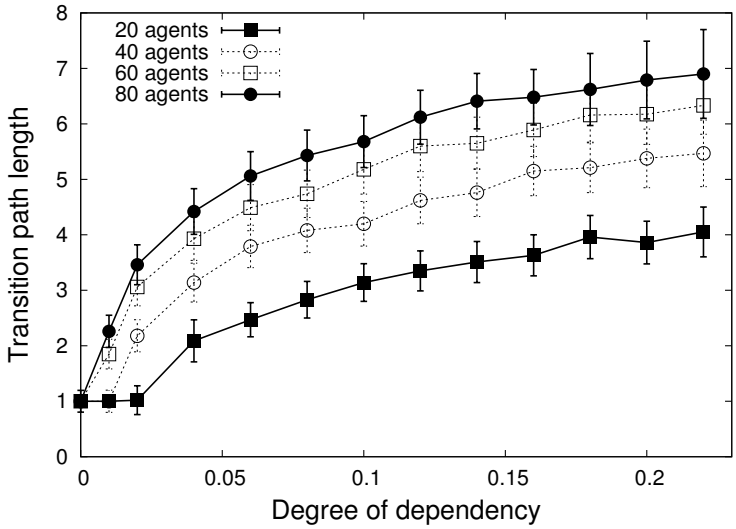

Figure 6.5: Transition path length for different agent population

Finally, the objective of the following test is to measure how often can be the organization deliberation for transitioning, depending on the population size and the services demand. In this experiment different agent populations are tested for five different service demand configurations. These configurations range between a configuration in which all the services are equally demanded (demand rate of 1) and a configuration in which the most demanded services are 5 times higher than the less demanded services (demand rate of 5). For each configuration, 100 time-steps are simulated and the number of time-steps between two consecutive transitions is shown in Figure 6.6. This number represents how long can be the interval for transition deliberation.

It can be observed that the demand rate influences the interval length for every agent population. The figure shows that when the demand for each service is similar, a shorter interval is required, and therefore, the frequency for transition deliberation is required to be higher. This can be explained due to the possibility of reallocating

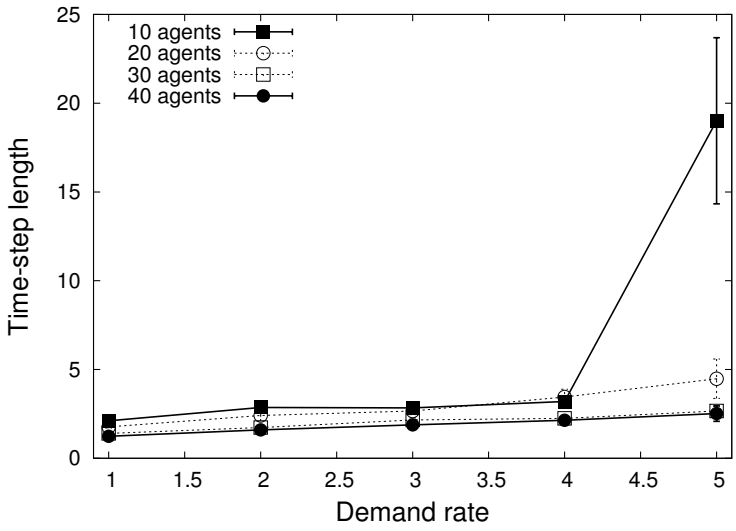

Figure 6.6: Interval length for transition deliberation

different agents to play roles becomes also higher. In contrast, when the demand of services is more different, this interval is higher. As an example, for a population of 10 agents, the interval is around 20. This means that deliberations may occur every 20 time-steps due to the roles do not require to be reallocated so often. It can also be observed in the figure the influence of the population size in the interval length. Note that as the population size increases, the interval length becomes shorter. This reflects that as the more agents can be reallocated to play each role, the frequency of transition deliberation should be higher, due to the range of reallocation possibilities is also higher.

6.5 Execution

In this section, we present a comparison of different approaches for organization transitions during an execution of several time-steps. The organization manager deliberates on whether or not to transition to another organization based on the benefits

of this organization and the transition costs. We carry out different experiments in order to show the capacity of the approach for achieving organizations that provide better utility by carrying out transitions. We also want to demonstrate the importance of considering costs before carrying out a transition since an organization that provides higher utility may not be worth the reorganization costs. Finally, we want to show the flexibility of the system for introducing new considerations in the organization transitions. To perform these tests, we introduce the possibility of transitioning to organizations in which the number of agents that play each role could be changed.

The organization is always composed of a set of twelve agents $a_0, \ldots, a_{11}$. Each agent has an associated suitability for providing each service which is determined to be between 0 and 1. Each agent plays a different role $r_0, \ldots, r_5$ according to the specific kind of tourist management service that each agent provides. At the initial moment t^0, each role is played by two agents that are randomly selected, without taking into account the suitability of each agent for providing each service. This random distribution is carried out in order to observe the adaptation ability of the organization.

For each test, we reproduce a system that simulates user agent requests for travel packages during the 50 time-steps. At t^0, the organization receives 30 requests from user agents for each one of the tourist services. As we stated in Section 6.1, the 30 requests of each service s_y are received by the two agents that play the role r_y. Then, in a period of time between two consecutive time steps, the number of requests for each service s_y are generated according to the following formula:

$$R(A, s_y)_{t-1}^{t} = R(A, s_y)_{t-2}^{t-1} \times random(0.95, 1.05)$$

As can be observed, the number of requests received in two consecutive time steps may change in an interval of $\pm$ 5%. However, in a real scenario, demand of services may change at any time. Therefore, in order to simulate a more realistic and dynamic execution, we varied the demand of services during the 50 time-steps. This causes

that some services are more or less demanded than others. These new requirements should cause the organization to transition to other organizations in order to provide better utility.

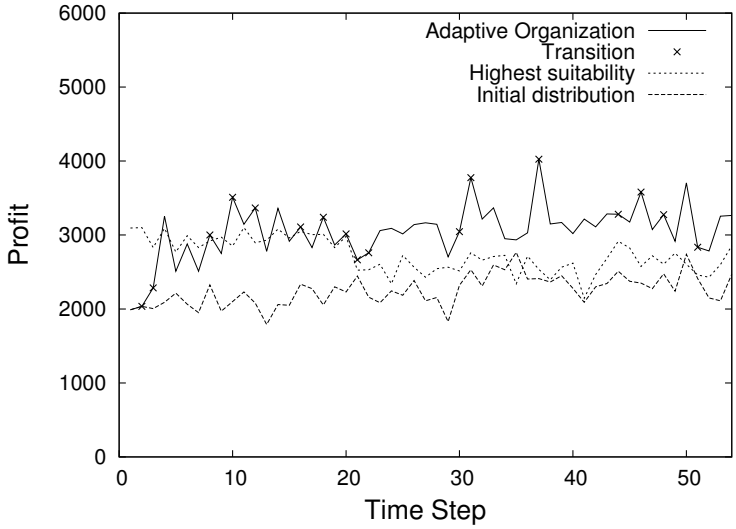

Figure 6.7: Static and dynamic organizations

Figure 6.7 shows a comparison of the utility of two static and one adaptive organization. Adaptive organization represents an organization that is able to carry out transitions. The figure also shows the time-steps in which transitions are carried out for this organization. After carrying out a transition, the transition cost in terms of the penalization for returning the unsold tourist packages is also reflected in the utility of the next time-step. The static organizations represent the initial distribution of services between roles, and a distribution of roles between a pair of agents that provide the service associated to the role with the highest suitability. This distribution tries to maximize the mean suitability provided by each service. However, the two best agents for providing a service may not play the role that offers this service if another role is played by two agents with low suitability.

Figure 6.7 shows that although the initial distribution of services in the adaptive organization is worse than the highest suitability distribution, the organization carries

out transitions in t^2 and t^3 in order to find an organization that makes higher profits. This increases the utility of the initial configuration until a utility similar to the highest utility is achieved. At t^{20}, the service demand by user agents changes from this moment on. This causes the demand for services s_2 and s_5 to increase while the demand for rest of the services decreases. As can be observed, the utility of the static organization with the highest suitability and the utility of the adaptive organization decrease at t^{21}. However, the adaptive organization carries out transitions in t^{21} and t^{22} in order to improve the utility. These transitions achieve organizations in which the pair of agents that plays roles r_2 and r_5 are those that provide s_2 and s_5 with the highest suitability, although other roles are played by agents with low suitability. When the demand for services is not similar to the initial demand, static organizations are not able to improve utility.

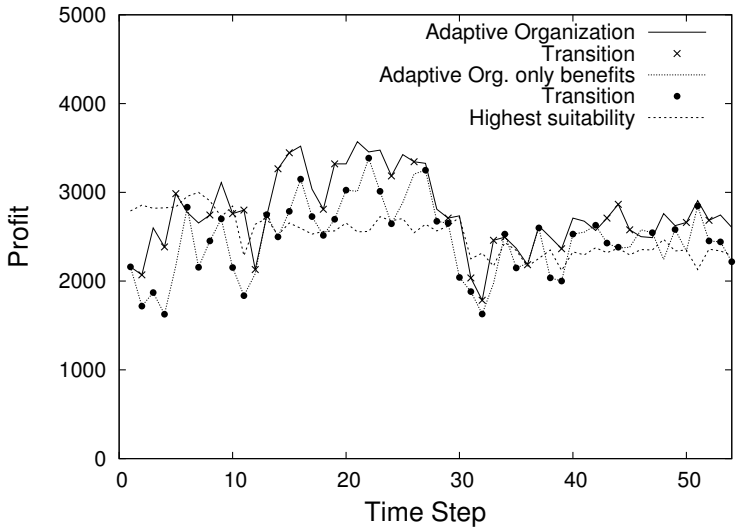

Figure 6.8: Transitions with costs and without costs

In order to measure the influence of considering costs for organization transitions, Figure 6.8 shows a comparison of the organization utility when costs for transition (i.e. the costs for returning the unsold requests) are taken into account and when they are not. When costs are not taken into account, the organization to be transitioned

to is the one with the highest expectation of increase in profits, independently of the costs required to achieve this organization. As the figure shows, in both cases adaptive organizations tend to improve the organization's utility. From t^0 on, several transitions are carried out in order to achieve a distribution of services that provides higher utility. Note that the adaptive organizations achieves a utility that is similar to the static organization with the highest suitability. Similar to the previous experiment, the service demand changes throughout the 50 time-steps. The service s_0 is the most demanded one from t^{10} on, while the service s_2 is the most demanded one from t^{30} on.

Adaptive organizations respond to these changes in demand. They generate organizations with provide better utility than the organization with highest suitability distribution. However, the utility of the adaptive organization that only considers the benefits generated is usually lower than the utility of the organization that also considers costs for transition, i.e., the utility is expected to increase; however, these transitions may not worth the cost. As an example, at t^6 the utility generated by the adaptive organization that only considers benefits is slightly better than the organization that also considers costs. Nevertheless, the costs required to carry out the transitions at t^2, t^3, and t^4 are not worth the profits generated. It is also important to note the great number of transitions carried out in the organization that does not consider costs. This great number of transitions also affects the lower utility of the organization that only considers benefits in comparison with the organization that also considers costs.

The following experiment introduces another requirement for consideration in organization transitions: the number of agents that play a role. Until now, the number of agents that play each role remains constant during the 50 time-steps. Organization transitions considers a role swap between two or more agents. In this experiment, the organization not only considers transitions to organizations where each role is played by a pair of agents, it also considers organizations in which the number of agents that play a role can be changed, i.e., a role cardinality transition. To deal with this problem, in each time-step, the organization considers several final organizations to

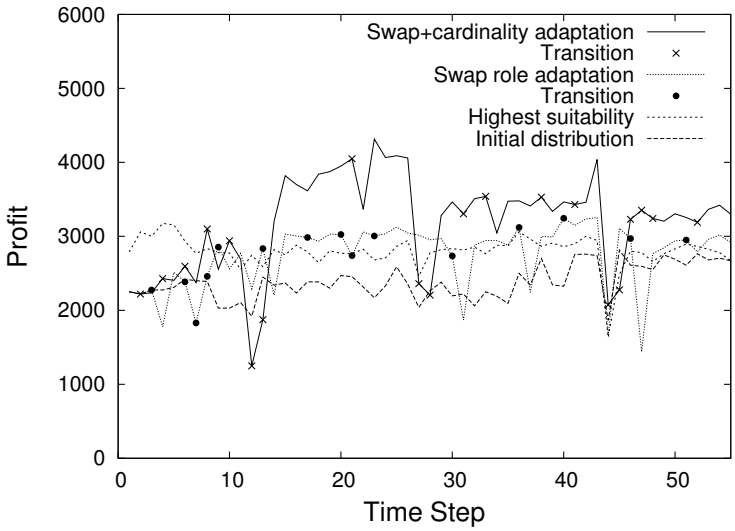

Figure 6.9: Transitions by swap role and role cardinality

be transitioned to by mantaining, increasing, or decreasing the number of agents that play each role. With this new transition consideration, the range of organizations that can be transitioned increases.

Figure 6.9 shows the utility of two adaptive organizations. One of these adaptive organizations considers role reallocation transition by role swap. The other adaptive organization also considers changes in the role cardinality. This figure also shows the utility of two static organizations: one of them with the same initial distribution as the adaptive organizations, and the other one with the higher suitability distribution. Note that, similar to the above experiments, adaptive organizations try to improve the utility of the organization, which increases up to t^{10}. From t^{10} on, the demand for the service s_0 becomes the most demanded one. At that moment, the performance of adaptive organizations changes abruptly; however, as in the above experiments, after carrying out several transitions, the utility of adaptive organizations is even better at t^{15} than the utility of the static organization with the highest suitability distribution. When service demand changes, the adaptive organization that considers role swap determines that the best distribution of services is to associate the role r_0 with the

brokers that provide the service s_0 with the highest utility. The adaptive organization that also considers role cardinality determines that it could be even better to increase the number of agents playing role r_0. In both adaptive organizations, the utility is better than the static organizations, which are not able to improve their utility when the demand for services changes. However, since the range of transitions considered by the adaptive organization that considers both swap and cardinality is greater, the utility is much better than the one that only considers a role swap. At t^{26} the demand for tourist services changes with the service s_5 being the most demanded one. Similar to previous change at t^{10}, changes in the demand may abruptly reduce the utility of the adaptive organizations. These organizations were adapted to provide high utility when service s_0 was the most demanded one. Nevertheless, adaptive organizations are able to carry out transitions in order to improve this performance. At t^{42}, the user demand changes again, where the services s_1 and s_2 are the most demanded ones and the behaviour of adaptive organizations is similar.

In summary, introducing new considerations in the computation of organization transitions increases the range of solutions that can be found. Therefore, the organization utility could also improve if other considerations are also taken into account.

6.6 Conclusions

In this chapter we evaluated the efficiency, the scalability, and the performance of the RF service. First, the execution trace showed the steps required for interaction between the organization manager and the RF service in order to carry out the transition. The experiments that have been carried out in Section 6.4, tested the RF service and the underlying algorithms for different organizational configurations (population, roles, services, etc.). Finally, we demonstrated different contributions of the proposed model for the role reallocation transition. We have shown the importance of considering costs before carrying out the transition. If costs are not accurately estimated, the benefits caused by reorganization may not be worth the costs required to carry

out the process. Moreover, if reorganization effects are not measured for the entire organization, an expected improvement in the performance of some specific agents might lead to a decrease in the performance of other agents.

As we have shown in the last experiment, by introducing new considerations for organization transitions such as the number of agents playing a role, the range of final organizations that could be transitioned to is increased. Thus, an organization transition with better expectations can be found if new considerations are taken into account. This last point is specially important in order to introduce the multi-dimensional transition requirement. If several dimensions are considered for transitions, the range of solutions would be increase, by achieving event better organizational behavior. The next chapter is focused on this issue. We show another case study which is focused on evaluating the features of the MTDM, specially the multi-dimensional transition support.

CHAPTER

7 Organization Transitions on a Service Provider Network

7.1 Introduction 129
7.2 Evaluation 137
7.3 Conclusions 143

The case study presented in this chapter is focused on testing the effectiveness of the MTDM by comparing the organization performance under different conditions of adaptation. This example is based on a network of data servers. After describing the case of study, we define the impact measurement for the three organization transitions provided by the MTDM. Then, a set of experiments is presented in order to demonstrate the contributions of the three principles of the MTDM: organization adaptation, cost-aware computation, and multi-dimensional transitions.

7.1 Introduction

Service Provider Networks (SPNs) represent environments for modeling agents that provide services to other agents. We model a SPN as an organization that is composed of agents that play different roles according to the services that they offer.

Each service involves a different kind of data resources, and external agents of the organization request these resources in order to retrieve them. In this example, we assume that each service s_y is provided by the role r_y.

Each agent of the organization is directly connected to other agents through bidirectional links. This is represented in the organization by acquaintance relationships. A single acquaintance (a_x, a_y) or (a_y, a_x) is sufficient to determine that agents a_x and a_y are directly connected. We represent by $L(a_x)$, the acquaintances agent a_x at the current moment: $L(a_x) = \{(i, j) \in acquaintance^c : i = a_x \vee j = a_x\}$. Depending on the number of acquaintances, an agent a_x has associated an specific bandwith BW_{a_x} that is computed as the division of a global bandwith BW between the number of acquaintances of a_x: $BW_{a_x} = \frac{BW}{|L(a_x)|}$. The global bandwith BW in this example is defined as constant for every agent.

A link between a pair of agents a_x and a_z, which are directly connected, has an associated bit-rate $BR(a_x, a_z)$, which represents the transfer speed when data resources are transferred through this link. This bit-rate is computed as the minimal of the bandwith of both agents: $BR(a_x, a_z) = min(BW_{a_x}, BW_{a_z})$. When agents are asked for a data resource that belongs to a service that is not provided by them, they must retrieve this data resource from any other agent of the organization by requiring a time for retrieving this data. There exists a path that connects any pair of agents a_x and a_z. The path that has the minimal number of links between a pair of agents, is the path through which all the transferences between these agents are carried out.

Each agent a_x receives a number of requests from agents for a service s_y in a time period between t' and t, which is represented as $R(a_x, s_y)^t_{t'}$. If the agent provides the requested service, it can send the data resource immediately, making the time required to have the data available to be null: $AT(a_x, R(a_x, s_y)^t_{t'}, s_y) = 0$.

If the agent does not provide the requested service, it must retrieve the data resource from another agent of the organization through its nearest provider $NP(a_x, s_y)$. This agent is the agent that is directly connected to a_x, which defines a path that has associated to it the minimal number of links for accessing an agent provider of service

s_y. Thus, the whole time required by a_x to have the data available can be calculated as $AT(a_x, R(a_x, s_y)^t_{t'}, s_y)$:

$$\frac{R(a_x, s_y)^t_{t'} \times Avg.(s_y)}{BR(a_x, NP(a_x, s_y))} + AT(NP(a_x, s_y), R(a_x, s_y)^t_{t'}, s_y) \qquad (7.1)$$

being $Avg(s_y)$ the average size of an individual data resource of the service s_y. The first addend of Eq. 7.1 corresponds to the time required for a_x to obtain the data from its nearest provider $NP(a_x, s_y)$, while the second addend corresponds to the time required for this agent for, in turn, obtain the data requested. Depending on the acquaintances between agents, a higher or a lower number of internal requests would be required.

In a SPN environment, data is usually transferred through links in both directions. Therefore, the transference time for each link between two agents a_x and a_z represents the whole time required for transfering all the data demanded through this link in a period of time between t' and t, i.e., the requests for each service retrieved from a_x and from a_z during this period of time that require this link. This transference time is represented as $T(a_x, a_z)^t_{t'}$:

$$\frac{\sum R(a_x, s_y)^t_{t'} \times Avg(s_y) + \sum R(a_z, s_w)^t_{t'} \times Avg(s_w)}{BR(a_x, a_z)} \qquad (7.2)$$

for all s_y, s_w such that $a_z = NP(a_x, s_y) \wedge a_x = NP(a_z, s_w)$.

The delay time of agent a_x represents the time required by a_x for retrieving all the data requested in a period of time. This is calculated as the maximum time required

by all the acquaintances of a_x for retrieving all the data requested between t' and t:

$$D(a_x)_{t'}^{t} = \max_{(i,j)\in L(a_x)} T(i,j)_{t'}^{t} \tag{7.3}$$

In this example, the utility of the organization can be represented as inversely proportional to the average delay time for retrieving all the requests received in the organization:

$$\bar{D}(O^t) = \frac{\sum_{a_x \in A} D(a_x)_{t'}^{t}}{|A|} \tag{7.4}$$

We used this notation to provide a consistent sense of utility measurement so that the system maximizes it, i.e. the system minimizes the average delay time for the whole organization.

The objective of the organization in this example is to minimize the time delay, i.e. to maximize the utility. At design time, it may be not posible to know what the best distribution of the services among the agents will be. Furthermore, according to the requests received, the best distribution may change while the system is running, and the organization must adapt to these new requirements. Therefore, organization transitions provide adaptation alternatives for improving this utility.

Following, we define how impacts are measured in this domain for the three types of organization transitions.

7.1.1 Role Reallocation Transition Impact Estimation

The impact of role reallocation for an agent a_x is obtained by Eq. 3.3. This equation calculates the impact of playing a new role an stop playing the current role. The term $I_A(a_x, r_n)$ (Eq. 3.1) represents the impact for agent a_x of providing the services offered by the new role r_n. In this example, since a single service is provided by a role, this impact refers to the time required to transfer the whole database for the service s_n to the agent a_x from the nearest provider of a_x.

According to Eq. 7.1, this time can be calculated as the availability time required for retrieving this database:

$$I_A(a_x, r_n) = i(add_provides((a_x, s_n))) = AT(a_x, N(s_n), s_n)$$

being $N(s_y)$ the number of files of the database provided by the service s_y. The term $i(add_plays((a_x, r_n)))$ represents the impact of playing the new role r_n once the new database has been transferred. This impact measures how the average delay time would be affected if a_x plays the role r_n. Since the number of requests that will be received from time-step t on is unknown, this number can be estimated according to the requests that have been received between t' and t for evaluating this impact. Thus, this impact can be represented as the negative time associated to the time gained by the requests that would not be transferred throughout the network if a_x plays this role:

$$i(add_plays((a_x, r_n))) = -AT(a_x, R(a_x, s_n)_{t'}^{t}, s_n)$$

The impact $I_D(a_x, r_c)$ (Eq. 3.2) represents the time required for deleting the database of the service provided by the current role r_c from agent a_x in a SPN environment. We assume this impact to be null because the time required for this operation can be minimal.

The term $i(delete_plays((a_x, r_c)))$ represents the impact of stopping a_x from playing the role r_c once the current database is deleted. This impact measures how the

average delay time would be affected if a_x does not play the role r_c anymore. Similar to the impact $i(add_plays((a_x, r_n)))$, this impact can be estimated as the time that would be required to provide the services requested between t' and t if agent a_x had not been playing r_c. These requests would be retrieved from the nearest provider of a_x:

$$i(delete_plays((a_x, r_c))) = AT(a_x, R(a_x, s_c)_{t'}^{t}, s_c)$$

7.1.2 Acquaintance Transition Impact Estimation

In a SPN environment, an acquaintance addition between a pair of agents causes the time required for transfering any data between these agents to decrease, since fewer links are required. This also may influence the nearest providers of all the agents of the organization. However, as stated in Section 7.1, the addition of a link between agents a_x and a_z causes the bandwith for any link of these agents to be reduced to $BW_{a_x} \times \frac{|L(a_x)|}{|L(a_x)|+1}$ for a_x and $BW_{a_z} \times \frac{|L(a_z)|}{|L(a_z)|+1}$ for a_z.

Similarly, deleting a link between agent a_x and a_z causes the bandwith for any link of these agents to be increased to $BW_{a_x} \times \frac{|L(a_x)|}{|L(a_x)|-1}$ for a_x and $BW_{a_z} \times \frac{|L(a_z)|}{|L(a_z)|-1}$ for a_z.

These bandwith modifications may cause the bit-rate of links that include one of these agents to be changed to $BR(i, j) = min(BW_i, BW_j)$, for all $(i, j) \in acquaintances^t$ such that $i = \{a_x, a_z\} \vee j = \{a_x, a_z\}$.

Similarly to the role reallocation transition estimation, the impact of adding and deleting acquaintances between agents can be estimated according to the requests that have been received between t' and t. Since a modification of an acquaintance may influence the nearest providers of other agents, requests that would be received at each agent can be recalculated according to this modification.

This causes that a new delay is calculated by following Eq. 7.3 and a new average delay time $\bar{D}_N(O^t)$ is estimated according to Eq. 7.4.

In order to compute the impact of adding an acquaintance between a pair of agents a_x and a_z, the difference between the average delay time of the current organization and the average delay time of the new organization is calculated. Moreover, a fixed cost associated to the time required for setting up the link is represented as $c_s(a_x, a_z)$.

Thus, the impact for adding an acquaintance between this pair of agents is calculated as:

$$i(add_acquaintance((a_x, a_z))) = \bar{D}_N(O^t) - \bar{D}(O^t) + c_s(a_x, a_z)$$

Similarly, a fixed cost associated to the time required for turning off a link between a pair of agents a_x and a_z is represented as $c_t(a_x, a_z)$. Thus, if two agents a_x and a_z are connected by an acquaintance, we can calculate the impact for deleting this acquaintance as $i(delete_acquaintance((a_x, a_z)))$:

$$\bar{D}_N(O^t) - \bar{D}(O^t) + c_t(a_x, a_z)$$

7.1.3 Agent Population Transition Estimation

In a SPN environment, the entrance or exit of an agent from the system has an impact on the delay of the whole organization. The impact related to the addition of an agent a_x is represented as $I_S(a_x)$ by following Eq. 3.6. This impact involves a fixed impact $i(add_agent(a_x))$ that represents the time required for setting up the new agent as $c_s(a_x)$. Moreover, the agent must be directly connected to another agent a_z in order to join the SPN with a fixed time for setting up the link $c_s(a_x, a_z)$ as stated in Section 7.1.2. This causes the bit-rate of links to be recalculated as stated in Section 7.1.2. Finally, the agent must play a specific role r_n that requires an impact of $I_S(a_x, r_n))$ that can be calculated by following Eq. 3.1. This impact is composed by $I_A(a_x, r_n)$ and $i(add_plays((a_x, r_n)))$. As stated in Section 7.1.1, the impact $I_A(a_x, r_n)$ can be computed as the time for retrieving the specific database $AT(a_x, N(s_n), s_n)$. Since agent a_x did not receive any previous request, we compute the impact $i(add_plays((a_x, r_n)))$ as null. We assume that all the requests received

in the organization during t' and t would have been distributed between this new population of agents. Then, the requests that would be received at every agent with this new configuration and the average delay time for this new organization $\bar{D}_N(O^t)$ can be calculated by following Eq. 7.4. Thus, the impact $I_S(a_x)$ can be calculated as:

$$\bar{D}_N(O^t) - \bar{D}(O^t) + c_s(a_x) + c_s(a_x, a_z) + AT(a_x, N(s_n), s_n)$$

The impact related to the deletion of an agent a_x is represented as $I_P(a_x)$ by following Eq. 3.7. This impact involves a fixed impact $i(delete_agent(a_x))$ that represents the time required for turning off the agent as $c_t(a_x)$. The deletion of all the links require a fixed impact for turning off these links that can be represented as $c_t(a_x, L) = \sum c_t(i, j)$ for all $(i, j) \in L(a_x)$. This also causes the bit-rate of acquaintances to be recalculated.

Finally, the impact for agent a_x to stop playing the role r_c is represented as $I_P(a_x, r_c)$ and can be calculated by following Eq. 3.2. This impact is composed by $I_D(a_x, r_c)$ and $i(delete_plays((a_x, r_c)))$. As stated in Section 7.1.1, the impact $I_D(a_x, r_c)$ can be considered null, while the impact $i(delete_plays((a_x, r_c)))$ can be estimated as $AT(a_x, R(a_x, s_c)^t_{t'}, s_c)$.

Similar to the addition operation, we assume that requests received at agent a_x would have been distributed between the rest of the agents of the organization if a_x is deleted.

Then, the requests that would be received at every agent and the average delay time for this new organization $\bar{D}_N(O^t)$ can be calculated by following Eq. 7.4. Thus, we can calculate the impact $I_P(a_x)$ as:

$$\bar{D}_N(O^t) - \bar{D}(O^t) + c_t(a_x) + c_t(a_x, L) + AT(a_x, R(a_x, s_c)^t_{t'}, s_c)$$

7.2 Evaluation

To show the performance of the MDTM, in this Section, we compare the average delay time for the organization (in seconds) on a SPN during 50 time-steps, when different approaches for transitions are applied. The organization at the initial moment t^0 is composed of a set of ten agents $\{a_0, \ldots, a_9\}$. Each agent plays a different $\{r_0, \ldots, r_3\}$ according to the service that each agent provides. At t^0 the distribution of roles are three agents playing r_0 and $_1$, and two agents playing r_2 and r_3. Each agent is connected to two agents by defining a ring topology. At t^0 the organization receives 100 requests that are distributed among all the agents: 25% of these requests are for s_0, 40% are for s_1, 20% are for s_2, and 15% are for s_3. Then, in a period of time between two consecutive time steps, each agent a_x receives requests for each service s_y according to the following formula:

$$R(a_x, s_y)_{t-1}^{t} = R(a_x, s_y)_{t-2}^{t-1} \times random(0.95, 1.05))$$

The number of requests received in two consecutive time steps may change in an interval of $\pm$ 5% ([0.05,-0.05]). However, these intervals may change in some agents for some services during the 50 time-steps in order to reproduce a dynamic scenario in which demand of specific services changes during one execution. For fixed costs for acquaintance and agent population transitions, we apply a cost of 10 seconds for setting or turning off an acquaintance, and a cost of 40 seconds, for setting or turning off an agent.

Figure 7.1 shows the average delay time for approaches that consider one-dimensional transitions and static organizations that do not consider transitions. The static organizations are regarding the initial configuration t^0 and a configuration that has the best performance from t^{27} to t^{34} (Static org. at t=27). The figure also shows the time-steps in which transitions are carried out. In these time-steps, the time required to carry out the transition is also reflected in the figure. Note that approaches that consider transitions adapt the organization according to the changes in the demand

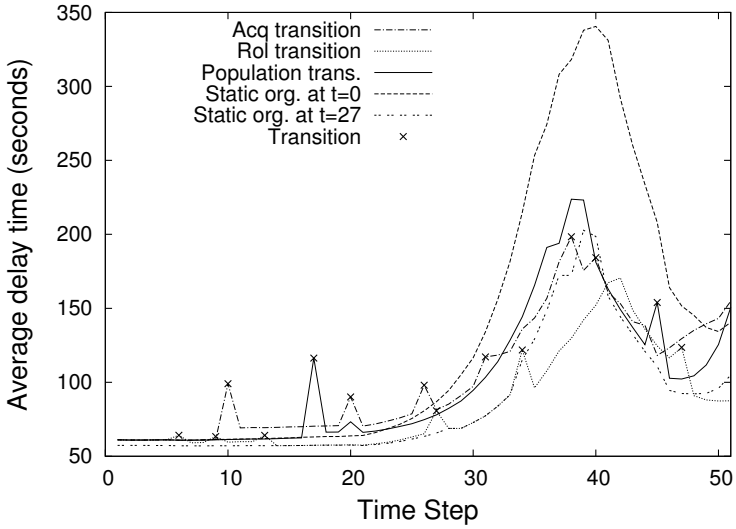

Figure 7.1: Static and dynamic organizations

of services. Although some static organizations may perform better at a specific moment than organizations with transitions, it can be observed that when circumstances change, static organizations are not able to respond to these changes and, thus, the average delay time may get worse. In this example, demand of service s_1 has been high from t^{21} on, while this demand has decreased from t^{37} on. Thus, the role reallocation transition approach achieves an organization at t^{27} in which the number of agents playing s_1 increases. Similarly, the acquaintance transition approach creates acquaintances to improve the average delay time. However, static organizations do not respond to these changes, which are unknown at design time and the performance gets worse. Table 7.1 shows the mean average delay time of this execution and 40 executions for a 95% confidence interval. It can be observed that in the current execution, the average delay time for the static organization configured as t^{27} is better than other approaches that take transitions into account. This is caused because the role distribution in this static organization is very good for this execution, while the acquaintance and the agent population approaches cannot change this distribution. Thus, the average delay time cannot be considerably improved by changing only

acquaintances or the agent population. However, the role reallocation transition approach, which can change this distribution, clearly outperforms the other approaches. As can be observed in the table, several executions causes that the performance of static organizations get worse because these approaches do not adapt to the demand of services. In this example can be observed that the role distribution is very influent in the performance.

Table 7.1: Static and dynamic organizations

	Average delay time (seconds)	
Approach	Execution	40 executions
Acq. approach	101.97	98.56±7.12
Role approach	84.65	82.75±6.24
Pop. approach	101.16	104.3±8.15
Static org at t^0	131.01	125.41±13.21
Static org at t^{27}	88.5	110.64±12.85

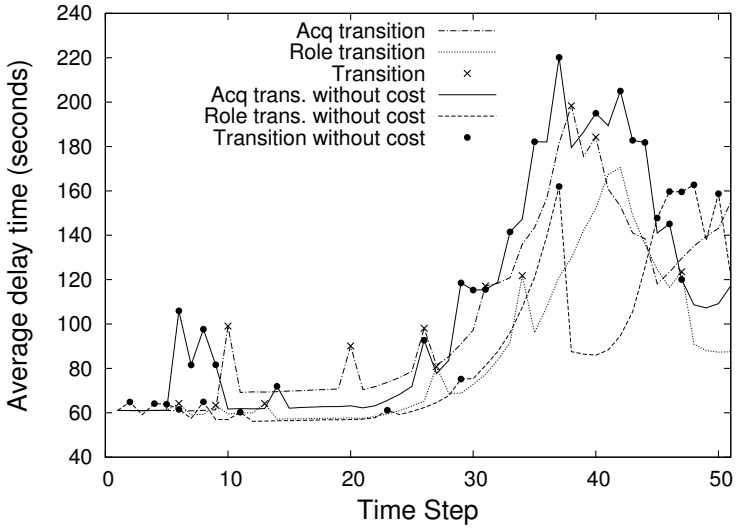

Figure 7.2: Transitions with costs and without costs

In order to measure the influence of adaptation costs, Figure 7.2 shows a comparison

between the average delay time for approaches that consider one-dimensional transitions of role reallocation and acquaintance, and transitions without considering the transition costs for deliberation. In a SPN environment, this corresponds to not considering the costs for retrieving the databases and the costs for setting and turning off acquaintances. As can be observed, approaches that adapt the organization without computing transition costs may transition to organizations with a great improvement in average delay time but at a high cost. These transitions may not be worth the cost, which causes the transition not to worh. The great number of transitions carried out in approaches that do not consider costs can also be observed. Table 7.2 shows the mean average delay time for each approach. Approaches that consider costs and benefits (measured as impacts) for transition deliberation, the mean average delay time is improved.

Table 7.2: Transitions with and without costs

	Average delay time (seconds)	
Approach	Execution	40 executions
Acq. approach	101.97	98.56±7.12
Role approach	84.65	82.75±6.24
Acq. without costs	107.10	105.49±8.41
Role without costs	86.06	85.15±6.98

Finally, in order to evaluate the multi-transition approach, Figure 7.3 shows a comparison between the average delay time for approaches that consider single-dimensional transitions and the MTDM approach. In this experiment, at t^0, the organization receives 200 requests that are distributed among all the agents: 35% of these requests are for s_0, 35% are for s_1, 20% are for s_2, and 10% are for s_3. These percentages change during the 50 time-steps similarly than previous experiments in order change the demand of services.

As can be observed in the figure, considering several transitions makes the deliberation mechanism able to take changes for different elements, while single-dimensional

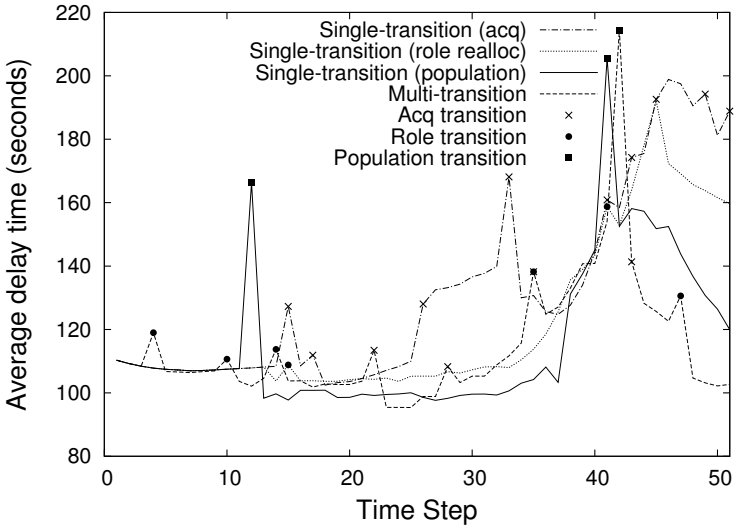

Figure 7.3: One-dimensional transitions and MTDM

Table 7.3: One-dimensional transitions and MTDM

	Average delay time (seconds)	
Approach	Execution	40 executions
Acq. approach	132.94	128.16±11.88
Role approach	122.27	124.62±10.77
Pop. approach	117.83	125.35±12.04
MTDM	114.24	112.2±6.59

approaches are limited to a specific kind of change, which might not always be the best adaptation possible. As an example, at t^{10} the transition achieved in the MTDM approach is by a role reallocation transition, while, at t^{28}, the transition is achieved by an acquaintance transition. Thus, the average delay time is improved by these two changes. Furthermore, since several changes can be carried out at the same time step, the possibilities of organization transitions are highly increased, and, thus, better transition decisions can be made. As an example, at t^{35}, the transition is carried out by a population transition and an acquaintance transition. Table 7.3, shows the mean

average delay time for each approach. It can be observed that the MTDM is able to find more possibilities for organizations to be transitioned to than one-dimensional transition approaches. Therefore, the organizations reached by the MTDM provide lower average delay time.

Table 7.4: Transitions effectiveness

	Percentage of transitions	
Type	Execution	40 executions
Acq. transition	36.36%	37.67±3.04%
Role transition	54.55%	53.96±3.29%
Pop. transition	9.09%	8.36±1.29%

In order to evaluate the effectiveness of the algorithm, Table 7.4 shows which percentage of transitions corresponds to each kind of changes. In this example, it can be observed that changes in the acquaintances and changes in the roles are the most frequent transitions. This behavior is caused because changes in the agent population also involves to add/delete acquaintances and to reallocate some roles. Therefore, depending on the cost of adding or deleting agents in the system, the percentage of agent population transitions could increase. What is more, depending on the costs (or penalization) of acquaintance modifications and role modifications, a greater or a lower number of these transitions would be carried out. As stated in Section 7.2, setting and turning off an agent has a higher cost than setting or turning of an acquaintance. However, the three kinds of transitions are decided at different moments of each individual execution. In summary, we can conclude that the requirement of considering several dimensions for transition it is beneficial due to different kinds of changes are carried out during each execution.

Finally, Table 7.5 shows the average execution time required by the algorithm during 50 time-steps depending on the initial agent population. It can be observed the increase in the time computation according to the agent population increases. However, we can observe the few computation time required in comparison with the time gained by adaptation in this domain. Otherwise, the time required for executing the

Table 7.5: Algorithm Execution time

Agent population	Time (ms)
10	870.59±11.42
20	1045.68±14.36
30	1285.77±19.12
40	1512.86±25.05
50	1723.95±34.65

algorithm should be considered in the transition computation cost.

7.3 Conclusions

In this chapter we evaluated the effectiveness of the MTDM. First, as in the previous case study (6), we demonstrated the adaptation influence in order to improve the organization performance. This adaptability allows the organization to find the best organizational configuration depending on the changing circumstances. In addition, we demonstrated the importance of considering costs for achieving transitions. As we observed in our experiments, if these are not considered, the transition costs may not worth the benefits that are achieved by this transition.

The other feature that has been tested in this chapter is regarding the multi-dimensional transition support. As we observed in our experiments, approaches that consider one-dimensional transitions (roles, structural topology, population, etc.) offer a more limited range of solutions than the MTDM. Thus, in heterogeneous scenarios such as the SNP, in which several changes can affect the performance of the organization, a multi-transition criterion for deliberation would provide better decisions for adaptation.

Part V

Conclusions

CHAPTER

8

Conclusions and Future Work

8.1 Conclusions 147
8.2 Future work 149
8.3 Contributions 152

8.1 Conclusions

The work presented in this thesis is aimed at going a step further in the research area of reorganization in agent societies. The motivation of the reorganization approach presented was to contribute in the advancement in the support for developing next generation of open and dynamic applications based on agent societies.

The main contributions of this work can be viewed from different perspectives. The analysis of the most relevant works related to reorganization in agent societies determines the current state of the art of the area. In the last few years, some works have been appeared related to the reorganization issue. As far as we are concerned, the comprehensive review carried out in Chapter 2 presents a novel work for illustrating the advantages and limitations of current approaches. Some of the considerations presented as open challenges has been dealt with the reorganization approach presented

in this thesis.

We focus on providing an accurate impact estimation since we approach reorganization as a problem of finding the best transition to a future organization that can be achieved. We designed a transition model which allows to represent an accurate measurement of the costs associated to a reorganization process and how this process affects the members of the society. This support allows us to improve the deliberation process without carrying out the transition. This is specially important in dynamic scenarios, in which the benefits associated to a transition are not measured in long-term. The cost-aware feature of our reorganization approach measures the sensitive dependence on initial conditions and environmental data, therefore, the best moment for transition is triggered with an associate measurement of the adaptation consequences.

Related to the previous issue, the multi-dimensional consideration for transition improves the range of future organizations that can be achieved, which helps to improve the best transition solution. As far as we are concerned, considering only a specific type of change could be appropriate for specific domains, but could not be suitable for general adaptation solutions. Similar to human societies, the flexibility of a reorganization process that consider different dimensions to be changed would help to better understand the adaptation requirement in systems which are thought to be open. The MTDM presented in Chapter 4 takes into account not only changes in independent dimensions but also how these changes are inter-dependent to each other.

The implementation integrated in Magentix provides the support for dynamic agent organizations in order to use the reorganization model. We must remark that even the MTDM considers simultaneous dimensions to be transitioned, these are limited to changes in the roles and services, changes in the relationships between agents, and changes in the agent population. However, the modular implementation provided by the RF service would help to easily integrate other kind of changes that could be considered depending on the organization model, e.g. changes in the norms that regulate the system.

Finally, the comprehensive evaluation by means of the case studies helps to demonstrate the benefits of the above contributions. On the one hand, by means of the performance evaluation carried out in chapter 6, we tested the scalability and efficiency of the RF and the underlying algorithms with different organizational configurations. Then, the execution presented in this same chapter and specially in the chapter 7, demonstrated the effectiveness of the MTDM, which provides reorganization decisions that consider the impact of the reorganization in the three types of transitional dimensions.

8.2 Future work

In this section we summarize some of the most promising future challenges related to the field of reorganization and adaptation in agent societies. Some of them have been highlighted in Chapter 2 after the analysis of the current state of the art and could be viewed as an extension of this thesis, while other have arisen during the development of this thesis, as emerging research lines related to this field.

8.2.1 Dynamic Monitoring for Detecting an Adaptation Requirement

As we shown in Chapter 2, monitoring is essential in order to be able to detect undesirable behavior that needs to be corrected. Static mechanisms to detecting the need of reorganization require the information that is monitored to be known in advance at design time, making it difficult to develop dynamic applications which can adapt at run time. It is necessary to count on an adaptive approach which allows to overcome these monitoring limitations imposed by static designs. Thus, an adaptive approach should apply not only to the behavior and structure of the system, but also to the design of the monitoring system [85], specially when dealing with the management of complex systems over long periods of time.

It would be interesting for the next generation of adaptive agent societies to have support that allow the dynamic specification of the rules that trigger a reorganization as well as the relevant information that is monitored. Thus, depending on the changing requirements of the system, the information required can change throughout the agent society's life-span. Static mechanisms that do not consider changes regarding which information needs to be monitored may result useful in small application domains with a priori well known organizational structures, but they would not be suitable for large-scale or complex systems. A

s the number of agents in the society and their complexity grows, much more information is exchanged between agents. Most of this information could be not useful at every moment of the execution and only contributes to considerably increase the traffic in the system, specifically in approaches in which a middleware or centralizing entity is the responsible of reorganization deliberation or implementation. Therefore, an adaptive approach should apply not only to the behavior and structure of the system, but also to the design of the monitoring system [85], specially when dealing with the management of complex systems over long periods of time.

8.2.2 Increasing the Range of Multi-dimensional Transitions

As we observed in the experiments that evaluates the MTDM support (Chapter 7), the three types of transitions are usually considered during the organization life-span (unless a transition is strongly costly). This fact determines that the more transition possibilities are considered, the better reorganization solution can be found. In a similar way that considering three of the most typical types of changes in agent societies (role reallocation, acquaintances, and agent population) improves the performance of one-dimensional approaches, if we consider other transition dimensions, the range of possible transitions would be increase and therefore, a better transition could be found. One of these dimensions that would be interesting to be included in the model in the one related to normative changes. As we stated in Chapter 2, some reorganization approaches consider changes in the norms. As an example, a norm related

to acquaintances cardinality could be changed in order to increase or decrease the number of maximum connections in a SPN environment. These kind of transitions would allow to increase the range of deliberation possibilities.

8.2.3 Using Trust for Organizational Services Adaptation

Another branch of future work is focused on the use of trust for service selection and adaptation. In the literature we can find several works that use trust as the measure for representing the experience that is usefull for service selection, which captures the consumer expectation of a particular service. As an example, a consumer may give a low rating to a service provider, and then, another consumer may use this low rating in order to select the service that is requested. In addition, this low rating may refer to specific items such as the delivery period. Thus, depending on the importance of other consumers in this item, the low rating could be important or not.

Related to this issue, if we approach this problem in a organizational context, individual services are provided by organizational agents, which must collaborate and coordinate in order to provide services to external agents. The organization could use trust measures from outside providers, in order to adapt (select the services that are provided, compose multiple services, etc.) the services that are provided depending on the organizational restrictions (such as a limited number of services that can be provided, the services maintainance cost, the composition cost, etc.). As an example, if some consumers associated a low rating to a specific service, the organization may deliberate to not offer this service anymore, or to decrease the number of agents which provide this service.

This approach would be particularly interesting in competitive scenarios, in which several organizations are considered and therefore, consumer agents could select the same service in different organizations. Therefore, the goal of each organization is to improve the services offer (according to organizational restrictions) by using trust, in order to attract the consumers. In addition, in this multi-organizational environment,

organizations could collaborate in order to compose multiple services.

8.2.4 Smart grid and Adaptation

Smart grid infrastructures are envisioned to be highly dynamic, providing scenarios which require adaptability, prediction, and optimization in order to improve the performance and the utility (such as the energy usage) of static configurations. To address these issues, multiagent systems have been used as a paradigm for modeling and simulating smart grid infrastructures, by providing intelligent software agents to individual devices.

Modeling smart grid infrastructures by means of agent societies would facilitate the organization, interaction, and cooperation of heterogeneous agents that interact to each other in order to achieve goals. Nevertheless, a number of potential challenges remain still open, especially those related to adaptation: how smart grid systems could self-adjust in order to optimize the usage and storage of individual devices; how the system could adapt according to changes in the environment, which introduce new constraints in the system; how individual devices can adjust their preferences to thesechanges; how the system can learn from previous changes; how the system can self-reconfigure due to or in order to prevent failures or disturbances, etc.

For the next generation of truly smart grid technologies, adaptability, openness, and flexibility are essential features in order to allow systems to be capable to adapt themselves to gain advantage in their current environments. Therefore, another branch of future work would be related to the use of adaptive agent societies in smart grid environments.

8.3 Contributions

This section presents publications related to this PhD Thesis that are published to date, as well as participation in other publications. They are organized in publi-

cations related to this PhD Thesis that appear in journals referenced in the Journal Citation Reports, Thomson Reuters, or in conferences indexed in the Computing Research and Education Association of Australasia (CORE); publications related to this PhD Thesis that are published in other conferences and book chapters; and finally, participation in other publications.

8.3.1 Indexed publications

J. M. Alberola, V. Julian and A. García-Fornes. Multi-dimensional Adaptation in MAS Organizations. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics.* In Press, 2013. **3,080 (2011) Journal Citation Reports Q1**.

- In this work we presented the MTDM (Chapter 4) and the case-study which evaluates this mechanism (Chapter 7).

J. M. Alberola, V. Julian and A. García-Fornes. Challenges for Adaptation in Agent Societies. *Knowledge and Information Systems.* In Press, 2013. **2,225 (2011) Journal Citation Reports Q1**.

- In this work we analyzed the state of the art of adaptation by highlighting some of the most interesting open issues in this area (Chapter 2).

J. M. Alberola, V. Julian and A. García-Fornes. Using Cost-Aware Transitions for Reorganizing Multiagent Systems. *Engineering Applications of Artificial Intelligence.* Vol. 26, n. 1. 63-75, 2013. **1,665 (2011) Journal Citation Reports Q1**.

- In this work we presented the RF service (Chapter 5) and the case-study which evaluates this framework (Chapter 6).

J. M. Alberola, V. Julian and A. García-Fornes. A Cost-Based Transition Approach for Multiagent Systems Reorganization. *In Proceedings of the 10th International*

Conference on Autonomous Agents and Multiagent Systems (AAMAS11). 1221-1222, 2011. **CORE A**.

- In this work we presented a summary of the Organization Transition Model (Chapter 3).

J. M. Alberola. A Cost-Oriented Reorganization Reasoning for Multiagent Systems Organization Transitions. *In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS11).* 1349-1350, 2011. **CORE A**.

- In this work we summarize the main proposal of the Thesis.

J. M. Alberola, V. Julian and A. García-Fornes. Multi-dimensional Transition Deliberation for Organization Adaptation in Multiagent Systems. *In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS12).* 1379-1380, 2012. **CORE A**.

- In this work we present the Multi-dimensional Transition Mechanism (Chapter 4).

J. M. Alberola, V. Julian and A. García-Fornes. Open Issues in Multiagent System Reorganization. *In Highlights in Practical Applications of Agents and Multiagent Systems. 9th International Conference on Practical Applications of Agents and Multiagent Systems.* 151-158, 2011. **CORE C**

- In this work we presented a summary of the evaluation of current approaches for reorganization presented in Chapter 2.

8.3.2 Other conferences

J. M. Alberola, L. Búrdalo, V. Julian, A. Terrasa and A. García-Fornes. Dynamic Monitoring for Adapting Agent Organizations. *In Proceedings of the 3rd Interna-*

tional Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS12). 121-134, 2012.

J. M. Alberola, V. Julian and A. García-Fornes. Cost-Aware Reorganization Service for Multiagent Systems. *In Proceedings of the 2nd International Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS11 @ AAMAS11).* 60-74, 2011.

8.3.3 Book chapters

J. M. Alberola, V. Julian and A. García-Fornes. Cost-Aware Reorganization Service for Multiagent Systems. *In Advanced Agent Technology*. 442-456, 2011.

8.3.4 Participation in other publications

J. M. Alberola, J. M. Such, V. Botti, A. Espinosa and A. García-Fornes. A Scalable Multiagent Platform for Large Systems. *Computer Science and Information Systems.* In Press, 2013. **0,625 (2011) Journal Citation Reports Q3**.

J. M. Alberola and A. García-Fornes. Using a Case-Based Reasoning Approach for Trading in Sports Betting Markets. *Applied Intelligence.* In Press, 2012. **0,849 (2011) Journal Citation Reports Q3**.

J. M. Such, **J. M. Alberola**, A. Barella and A. García-Fornes. A Secure Group-Oriented Framework for Intelligent Virtual Environments. *Computing and Informatics.* 1225-1246, 2011. **0,239 (2011) Journal Citation Reports**. **Q4**

J. M. Such, **J. M. Alberola**, A. Espinosa and A. García-Fornes. A Group-oriented Secure Multiagent Platform. *Software: Practice and Experience.* 1289-1302, 2011.

0,519 (2011) Journal Citation Reports. **Q3**.

J. M. Alberola, J. M. Such, A. García-Fornes, A. Espinosa and V. Botti. A performance evaluation of three multiagent platforms. *Artificial Intelligence Review*. Vol. 34, num. 2. 145-176, 2010. **1,1213 (2011) Journal Citation Reports Q2**.

Ricard L. Fogués, **J. M. Alberola**, J. M. Such, A. Espinosa and A. García-Fornes. Towards Dynamic Agent Interaction Support in Open Multiagent Systems. *In Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*. Vol. 220, 89-98, 2010.

J. M. Alberola, A. García-Fornes and A. Espinosa. Price Prediction in Sports Betting Markets. *In Proceedings of the 8th German Conference on Multiagent System Technologies*. Vol. 6251, 197-208, 2010. **CORE B**.

J. M. Such, **J. M. Alberola**, A. García-Fornes, A. Espinosa and V. Botti. Kerberos-based Secure Multiagent Platform. *In Programming Multi-Agent Systems*. Vol. 5442, 197-210, 2009.

J. M. Alberola, J. M. Such, A. Barella and A. García-Fornes. Towards a Secure, Group-Oriented and Efficient Framework for Intelligent Virtual Environments. *In Artificial Intelligence Research and Development*. Vol. 202 pp. 234-241, 2009.

J. M. Such, **J. M. Alberola**, A. Barella, A. Espinosa and A. García-Fornes. A Secure Group-Oriented Model for Multiagent Systems. *In Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN2009) (Biological Inspired Systems. Computational and Ambient Intelligence)*. 522-529, 2009. **CORE B**.

J. M. Alberola, J. M. Such, A. Espinosa, V. Botti and A. García-Fornes. Magentix: a Multiagent Platform Integrated in Linux. *In Proceedings of the 6th European Workshop on Multiagent Systems.* 1-10, 2008. **CORE C**.

J. M. Alberola, J. M. Such, A. Espinosa, V. Botti and A. García-Fornes. Scalable and Efficient Multiagent Platform Closer to the Operating System. *In Artificial Intelligence Research and Development.* Vol. 184, 7-15, 2008.

J. M. Such, **J. M. Alberola**, A. García-Fornes, A. Espinosa and V. Botti. Kerberos-based Secure Multiagent Platform. *In Proceedings of the 6th International Workshop on Programming Multi-Agent Systems.* 173-186, 2008. **CORE B**.

J. M. Alberola, L. Mulet, J. M. Such, A. García-Fornes, A. Espinosa and V. Botti. Operating System Aware Multiagent Platform Design. *In Proceedings of 5th European Workshop On Multi-Agent Systems.* 658-667, 2007. **CORE C**.

J. M. Such, **J. M. Alberola**, L. Mulet, A. García-Fornes, A. Espinosa and V. Botti. Hacia el Diseño de Plataformas Multiagente Cercanas al Sistema Operativo. *In Proceedings of the International Workshop on Practical Applications of Agents and Multiagent Systems.* 1-10, 2007.

J. M. Such, **J. M. Alberola**, L. Mulet, A. Espinosa, A. García-Fornes and V. Botti. Large-Scale Multiagent Platform Benchmarks. *In Languages, Methodologies and Development Tools for MAS.* 192-204, 2007.

L. Mulet, **J. M. Alberola**, J. M. Such, V. Botti, A. Espinosa, A. García-Fornes and A.

Terrasa. Performance evaluation of open-source multiagent platforms. *In Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. Vol. 1 pp. 1107-1109, 2006. **CORE A**.

Bibliography

[1] AMQP: Advanced message queuing protocol, version 0.10. `http://www.amqp.org/resources/specifications`.

[2] A. Aamodt and E. Plaza. Case-based reasoning; Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[3] S. Abdallah and V. Lesser. Multiagent Reinforcement Learning and Self-Organization in a Network of Agents. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*, pages 172–179, 2007.

[4] H. Abdu, H. Lutfiyya, and M. A. Bauer. A model for adaptive monitoring configurations. In *Proceedings of the 6th IFIP/IEEE IM Conference on Network Management*, pages 371–384, 1999.

[5] J. M. Alberola, V. Julian, and A. Garcia-Fornes. A cost-based transition approach for multiagent systems reorganization. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS11)*, pages 1221–1222, 2011.

[6] J. M. Alberola, V. Julian, and A. Garcia-Fornes. Multi-dimensional Transition

Deliberation for Organization Adaptation in Multiagent Systems. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS12)*, pages 1379–1380, 2012.

[7] H. Aldewereld, F. Dignum, V. Dignum, and L. Penserini. A formal specification for organizational adaptation. In *Proceedings of the 10th International Workshop on Agent-Oriented Software Engineering (AOSE10)*, pages 18–31, 2011.

[8] E. Argente, V. Botti, C. Carrascosa, A. Giret, V. Julian, and M. Rebollo. An abstract architecture for virtual organizations: The THOMAS approach. *Knowledge Information Systems*, 29(2):379–403, 2011.

[9] E. Argente, V. Julian, and V. Botti. Multi-Agent System Development Based on Organizations. *Electronic Notes in Theoretical Computer Science*, 150(3):55–71, 2006.

[10] E. Argente, V. Julian, and V. Botti. MAS Modeling based on Organizations. In *Post-Proceedings of the 9th International Workshop on Agent-Oriented Software Engineering (AOSE08)*, volume 5386, pages 16–30. Springer, 2009.

[11] S. Ashford, R. Blatt, and D. Walle. Reflections on the Looking Glass: A Review of Research on Feedback-Seeking Behavior in Organizations. *Journal of Management*, 29(6):773–799, 2003.

[12] S. Ashford and M. Taylor. Adaptation to work transitions: An Integrative approach. *Research in personnel and human resource management*, 8:1–39, 1990.

[13] W. Astley and A. V. de Ven. Central Perspectives and Debates in Organization Theory. *Administrative Science Quarterly*, 28(2):245–273, 2003.

[14] K. M. Baum L.E. Convergence rates in the law of large numbers. *Transactions of the American Mathematical Society*, 120:108–123, 1965.

[15] A. H. Bond and L. Gasser. A survey of distributed artificial intelligence. In *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, 1988.

[16] E. Bou, M. López-Sánchez, and J. A. Rodríguez-Aguilar. Towards self-configuration in autonomic electronic institutions. In *COIN 2006 Workshops. Number LNAI 4386*, pages 220–235. Springer, 2007.

[17] E. Bou, M. López-Sánchez, and J. A. Rodríguez-Aguilar. Using case-based reasoning in autonomic electronic institutions. In *Proceedings of the 2007 International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems (COIN07)*, pages 125–138, 2008.

[18] E. Bou, M. López-sánchez, and J. A. Rodríguez-aguilar. Adaptation of autonomic electronic institutions through norms and institutional agents. In *Engineering Societies in the Agents World. Number LNAI 4457*, pages 300–319. Springer, 2006.

[19] J. Brett, D. Feldman, and L. Weingart. Feedback-seeking behavior of new hires and job changers. *Journal of Management*, 16:737–749, 1990.

[20] B. Bulka, M. Gaston, and M. desJardins. Local strategy learning in networked multi-agent team formation. *Autonomous Agents and Multi-Agent Systems*, 15(1):29–45, 2007.

[21] J. Campos, M. Esteva, M. Lopez-Sanchez, J. Morales, and M. Salamo. Organisational adaptation of multi-agent systems in a peer-to-peer scenario. *Computing*, 91:169–215, 2011.

[22] J. Campos, M. López-Sánchez, and M. Esteva. Assistance layer, a step forward in Multi-Agent Systems Coordination Support. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS09)*, pages 1301–1302, 2009.

[23] K. M. Carley and L. Gasser. Computational organization theory. In *Multiagent systems: a modern approach to distributed artificial Intelligence*, pages 299–330. MIT Press, 1999.

[24] G. Carvalho, H. Almeida, M. Gatti, G. Vinicius, R. Paes, A. Perkusich, and C. Lucena. Dynamic law evolution in governance mechanisms for open multi-agent systems. In *Proceedings of the Second Workshop on Software Engineering for Agent-oriented Systems (SEAS06)*, 2006.

[25] L. Cernuzzi and F. Zambonelli. Adaptive organizational changes in agent-oriented methodologies. *Knowledge Engineering Review*, 26(2):175–190, 2011.

[26] B. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, editors. *Software Engineering for Self-Adaptive Systems*. Springer-Verlag, Berlin, Heidelberg, 2009.

[27] D. Corkill and S. Lander. Diversity in Agent Organizations. *Object Magazine*, 8(4):41–47, 1998.

[28] D. D. Corkill and V. R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI83)*, pages 748–756, 1983.

[29] J. de Paz, J. Bajo, A. González, S. Rodríguez, and J. Corchado. Combining case-based reasoning systems and support vector regression to evaluate the atmosphere-ocean Interaction. *Knowledge Information Systems*, 30(1):155–177, 2012.

[30] S. DeLoach and E. Matson. An organizational model for designing adaptive multiagent systems. In *The AAAI-04 Workshop on Agent Organizations: Theory and Practice (AOTP)*, pages 66–73, 2004.

[31] S. DeLoach, W. Oyenan, and E. Matson. A capabilities-based model for adaptive organizations. *Autonomous Agents and Multi-Agent Systems*, 16:13–56, 2008.

[32] V. Dignum. *A model for organizational Interaction: based on agents, founded in logic*. PhD dissertation, Universiteit Utrecht. SIKS dissertation series 2004-1, 2004.

[33] V. Dignum, editor. *Multi-agent Systems: Semantics and Dynamics of Organizational Models*. IGI Global, 2009.

[34] V. Dignum and F. Dignum. Modelling agent societies: Co-ordination frameworks and institutions. In *Progress in Artificial Intelligence, LNAI 2258*, pages 191–204. Springer-Verlag, 2001.

[35] V. Dignum and F. Dignum. A Logic for Agent Organizations. In V. Dignum, editor, *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pages 220–240. Information Science Reference, 2009.

[36] V. Dignum, F. Dignum, and L. Sonenberg. Towards dynamic reorganization of agent societies. In *Proceedings of Workshop on Coordination in Emergent Agent Societies*, pages 22–27, 2004.

[37] V. Dignum, J. Vázquez-salceda, and F. Dignum. Exploring congruence between organizational structure and task performance: A simulation approach. pages 213–230, 2006.

[38] D. A. Dolgov and E. H. Durfee. Resource allocation among agents with mdp-induced preferences. *Journal of Artificial Intelligence Research*, 27:505–549, 2006.

[39] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[40] S. Esparcia and E. Argente. Formalizing Virtual Organizations. In *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART11)*, pages 84–93, 2011.

[41] R. L. Fogués, J. M. Alberola, J. M. Such, A. Espinosa, and A. Garcia-Fornes. Towards dynamic agent interaction support in open multiagent systems. In *Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence (CCIA10)*, volume 220, pages 89–98, 2010.

[42] M. S. Fox. An organizational view of distributed systems. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 11, pages 70–80. 1981.

[43] E. Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented Open Multiagent Systems. In *Proceedings of the 12th International Conference on Principles of Practice in Multi-Agent Systems (PRIMA09)*, pages 345–360, 2009.

[44] M. E. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the 4th International Joint Conference on Autonomous agents and Multiagent Systems (AAMAS05)*, pages 230–237, 2005.

[45] N. Glaser, V. Vouton, and P. O. Box. The reorganization of societies of autonomous agents. *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW97)*, pages 98–111, 1997.

[46] C. V. Goldman and J. S. Rosenschein. Evolving organizations of agents. In *Proceedings of the Workshop on Multiagent Learning at AAAI97*, pages 25–30, 1997.

[47] H. R. Greve. Performance, Aspirations, and Risky Organizational Change. *Administrative Science Quarterly*, 43(1):58–86, 1998.

[48] Z. Guessoum, M. Ziane, and N. Faci. Monitoring and organizational-level adaptation of multi-agent systems. In *Proceedings of the 3rd International*

Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS04), pages 514–521, 2004.

[49] M. Hoogendoorn. Adaptation of organizational models for multi-agent systems based on max flow networks. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI07)*, 2007.

[50] M. Hoogendoorn and J. Treur. An Adaptive Multi-agent Organization Model Based on Dynamic Role Allocation. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT06)*, pages 474–481, 2006.

[51] B. Horling, B. Benyo, and V. Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 529–536, 1999.

[52] B. Horling and V. Lesser. A Survey of Multi-Agent Organizational Paradigms. *Knowledge Engineering Review*, 19(4):281–316, 2005.

[53] L. G. Hrebiniak and W. F. Joyce. Organizational Adaptation: Strategic Choice and Environmental Determinism. *Administrative Science Quarterly*, 30(3):336–349, 1985.

[54] J. F. Hübner, O. Boissier, and J. S. Sichman. Specifying e-alliance contract dynamics through the moise + reorganisation process. In *Anais do V Encontro Nacional de Inteligencia Artificial (ENIA05)*, pages 434–443, 2005.

[55] J. F. Hübner, J. S. Sichman, and O. Boissier. Using the MOISE+ for a Cooperative Framework of MAS Reorganisation. In *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA04)*, pages 506–515, 2004.

[56] J. F. Hübner, J. S. a. Sichman, and O. Boissier. MOISE+: towards a structural, functional, and deontic model for MAS organization. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS02)*, pages 501–502, 2002.

[57] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44:35–41, 2001.

[58] V. Kachitvichyanukul and B. W. Schmeiser. Binomial random variate generation. *Communications of the ACM*, 31(2):216–222, 1988.

[59] S. Kamboj and K. S. Decker. Organizational self-design in semi-dynamic environments. In *Proceedings of the 2007 IJCAI workshop on Agent Organizations: Models and Simulations*, pages 335–337, 2006.

[60] D. Katz and R. Kahn. *The social psychology of organizations*. John Wiley & Sons, 1966.

[61] D. Kelly and T. L. Amburgey. Organizational Inertia and Momentum: A Dynamic Model of Strategic Change. 34(3):591–612, 1991.

[62] J. Kephart and D. Chess. The Vision of Autonomic Computing. *Computer*, 36(1):41–50, 2003.

[63] D. H. Kim. The link between individual and organizational learning. *Sloan Management Review*, 35(1):37–50, 1993.

[64] R. Kota, N. Gibbins, and N. R. Jennings. Decentralised structural adaptation in agent organisations. pages 54–71, 2008.

[65] R. Kota, N. Gibbins, and N. R. Jennings. Self-organising agent organisations. In *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS09)*, pages 797–804, 2009.

[66] R. Kota, N. Gibbins, and N. R. Jennings. Decentralised approaches for self-adaptation in agent organisations. pages 1–28, 2012.

[67] J. Kotter and L. Schlesinger. Choosing strategies for change. In *Harvard Business Review*, pages 106–114, 1979.

[68] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[69] B. Levitt and J. G. March. Organizational Learning. *Annual Review of Sociology*, 14:319–340, 1988.

[70] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. University of Southampton, 2005.

[71] P. Mathieu, J. C. Routier, and Y. Secq. Dynamic organization of multi-agent systems. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS02)*, pages 451–452, 2002.

[72] P. Mathieu, J.-C. Routier, and Y. Secq. Principles for dynamic multi-agent organizations. In *Proceedings of the 5th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems (PRIMA02)*, pages 109–122, 2002.

[73] E. Matson and R. Bhatnagar. Properties of capability based agent organization transition. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT06)*, pages 59–65, 2006.

[74] E. Matson and S. DeLoach. Using dynamic capability evaluation to organize a team of cooperative, autonomous robots. In *Proceedings of the 2003 International Conference on Artificial Intelligence (ICAI03)*, pages 23–26, 2003.

[75] E. Matson and S. DeLoach. Enabling intra-robotic capabilities adaptation using an organization-based multiagent system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2135–2140, 2004.

[76] E. Matson and S. DeLoach. Formal transition in agent organizations. In *Proceedings of the IEEE International Conference on Knowledge Intensive Multiagent Systems (KIMAS05)*, pages 18–21, 2005.

[77] H. Min. International supplier selection:: A multi-attribute utility approach. In *International Journal of Physical Distribution & Logistics Management*, pages 24–33, 1994.

[78] J. Morales, M. López-Sánchez, and M. Esteva. Using experience to generate new regulations. pages 307–312, 2011.

[79] D. Muhlestein and S. Lim. Online learning with social computing based Interest sharing. *Knowledge Information Systems*, 26(1):41–50, 2011.

[80] R. Nair and M. Tambe. Hybrid BDI-POMDP framework for multiagent teaming. *Journal of Artificial Intelligence Research*, 23:367–420, 2005.

[81] R. Nair, M. Tambe, and S. Marsella. Role allocation and reallocation in multiagent teams: towards a practical analysis. In *Proceedings of the Second International Joint Conference on Agents and Multiagent Systems (AAMAS03)*, pages 552–559, 2003.

[82] W. J. Orlikowski. Improvising Organizational Transformation Over Time: A Situated Change Perspective. *Information Systems Research*, 7(1):63–92, 1996.

[83] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, 2005.

[84] M. Rebollo, A. Giret, E. Argente, C. Carrascosa, J. Corchado, A. Fernandez, and V. Julian. On the road to an abstract architecture for open virtual organizations. In *Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN09)*, pages 642–650, 2009.

[85] P. Ringold, J. Alegria, R. Czaplewski, B. Mulder, T. Tolle, and K. Burnett. Adaptive Monitoring Design for Ecosystem Management. *Ecological Applications*, 6(3):745–747, 1996.

[86] J. Routier, P. Mathieu, and Y. Secq. Dynamic Skill Learning: A Support to Agent Evolution. pages 25–32, 2001.

[87] T. Schlegel and R. Kowalczyk. Towards self-organising agent-based resource allocation in a multi-server environment. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*, pages 74–81, 2007.

[88] M. Schwaninger. A theory for optimal organization. Technical report, Institute of Management at the University of St. Gallen, Switzerland, 2000.

[89] R. W. Scott. *Organizations: Rational, Natural, and Open Systems (5th Edition)*. Prentice Hall, 2002.

[90] A. Seelam. *Reorganization of massive multiagent systems: MOTL/O*. Southern Illinois University Carbondale, 2009.

[91] Y. So and E. H. Durfee. An organizational self-design model for organizational change. In *Proceedings of the AAAI93 Workshop on AI and Theories of Groups and Oranizations*, pages 8–15, 1993.

[92] Y. So and E. H. Durfee. *Designing organizations for computational agents*, pages 47–64. MIT Press, 1998.

[93] C. Tantipathananandh and T. Berger-Wolf. Finding Communities in Dynamic Social Networks. In *Proceedings of the IEEE 11th International Conference on Data Mining*, pages 1236–1241, 2011.

[94] N. Tinnemeier, M. Dastani, and J.-J. Meyer. Programming norm change. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS10)*, pages 957–964, 2010.

[95] J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing Multiagent Systems. *Autonomous Agents and Multi-Agent Systems*, 11:307–360, 2005.

[96] D. Wang, Q. Tse, and Y. Zhou. A decentralized search engine for dynamic Web communities. *Knowledge Information Systems*, 26(1):105–125, 2011.

[97] Z. G. Wang and X. H. Liang. A graph based simulation of reorganization in multi-agent systems. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT06)*, pages 129–132, 2006.

[98] K. Weick. *The social psychology of organizing*. Addison-Wesley, 1979.

[99] D. Weyns and M. Georgeff. Self-Adaptation Using Multiagent Systems. *IEEE Software*, 27(1):86–91, 2010.

[100] D. Weyns, R. Haesevoets, A. Helleboogh, T. Holvoet, and W. Joosen. The MACODO middleware for Context-Driven Dynamic Agent Organzations. *ACM Transactions on Autonomous and Adaptive Systems*, pages 3–28, 2010.

[101] D. Weyns, S. Malek, and J. Andersson. FORMS: a formal reference model for self-adaptation. In *Proceedings of the 7th International Conference on Autonomic Computing*, pages 205–214, 2010.

[102] C. Zhong. An investigation of reorganization algorithms. Master thesis, 2006.

[103] C. Zott. Dynamic capabilities and the emergence of intraindustry differential firm performance: Insights from a simulation study. *Strategic Management Journal*, 24(2):97–125, 2003.