Part I
Organisations and Institutions

Organisations and Institutions provide an interesting perspective for open Multi-Agent Systems and Agreement Technologies. For example, organisations can be employed to specify how to solve a complex task or problem by a number of agents in a declarative way; agents participating in an organisation can work together and form teams for the solution of a particular task that helps reaching the global goals of the organisation; organisational structures can improve and accelerate co-ordination processes in open environments. Moreover, the notion of institution has been used within the agent community to model and implement a variety of socio-technical systems, enabling and regulating the interaction among autonomous agents in order to ensure norm compliance.

This part addresses how agent organisationsand institutions can improve and accelerate coordination processes in open environments. A state-of-art of recent proposals for describing agent organisations is given in Chap. 1, relating the different methodologies and formal approaches for defining agent organisations in an explicit way. Moreover, a review and comparison of recent approaches of Artificial Institutions is provided in Chap. 2. Furthermore, there have been some recent approaches for developing agents capable of understanding the organisation structure and functionality and then being able for deciding whether participate inside or even generate new structures for the organisation. A review of this kind of agents, known as organisation-aware agents, is provided in Chap. 3. Finally, an important question in open systems is how to endow an organisation with autonomic capabilities to yield a dynamical answer to changing circumstances. Thus, a review of methods for designing and/or implementing adaptive agent organisations is given in Chap. 4.

Chapter 1
Describing agent organisations

Estefania Argente and Olivier Boissier and Sergio Esparcia and Jana Görmer and Kristi Kirikal and Kuldar Taveter

Abstract This chapter addresses how agent organisations can improve and accelerate coordination processes in open environments. A state-of-art of recent proposals for describing agent organisations is given, relating the different methodologies and formal approaches for defining agent organisations in an explicit way. As example, four different proposals developed within the COST action IC0801 are detailed: (i) the MOISE organisation Model, which provides structural, functional and normative specifications of an organisation, and it is integrated in an Organisation Management infrastructure; (ii) the Virtual Organisation Model, which describes the structural, functional, dynamical, environmental and normative dimensions of an organisation, and it is complemented by the Virtual Organisation Formalization; (iii) the Agent-Oriented Modelling for sociotechnical systems, which are organisations consisting of human and man-made agents; and (iv) the AAOL agent architecture, in which groups of autonomous agents are organized in Localities. This chapter proposes a global comparison of different organisational existing model with the four detailed models in terms of the different description dimensions they propose.

Estefania Argente
Universidad Politecnica de Valencia, Valencia, Spain, e-mail: `eargente@dsic.upv.es`

Olivier Boissier
ISCOD - LSTI ENS Mines Saint-Etienne, Saint-Etienne Cedex, France,e-mail: `Olivier.Boissier@emse.fr`

Sergio Esparcia
Departamento Sistemas Informaticos y Computacion, Universidad Politecnica de Valencia, Valencia, Spain e-mail: `serparcia@dsic.upv.es`

Jana Görmer
Department of Informatics, TU Clausthal, Clausthal, Germany e-mail: `jana.goermer@tu-clausthal.de`

Kristi Kirikal
Tallinn University of Technology, Tallinn, Estonia e-mail: `kristi.kirikal@gmail.com`

Kuldar Taveter
Tallinn University of Technology, Tallinn, Estonia e-mail: `kuldar.taveter@ttu.ee`

1.1 Introduction

To cope with the openness, decentralisation and dynamicity of applications targeted by Multi-Agent technologies, an organisational perspective has been promoted in the domain these last years. This perspective proposes that the joint activity inside Multi-Agent Systems should be explicitly regulated by a consistent body of norms, plans, mechanisms and/or structures formally specified to achieve some definite global purpose. Inspired by the metaphor with human organisations [56], different organisational models have been proposed in the literature, for the engineering of such systems (e.g. [24, 48, 41, 22, 18, 37, 60, 54, 57, 3]).

An organisational model consists of a conceptual framework and a syntax in which specifications for agent organisations can be written. We call this an Organisation Modelling Language (OML). From such specifications, called hereafter organisational specification, an organisation can be enacted on a traditional multi-agent platform or, more realistically, by using some organisation management infrastructure (OMI) [39, 42, 23, 34]. In general, these organisation management infrastructures take the organisational specifications as input, interpret them, and provide the agents with an organisation according to the given specification. In order to enter, to work inside or to leave the agent organisation, the agents are supposed to know how to access the services of the infrastructure and to make requests according to the available organisational specification. Equipped with such capabilities, agents develop what we call *Organisation Awareness* skills making them able to reason on the organisation to decide to enter or not in such a structure, to change it by setting in place a reorganisation process and finally to comply or not to the different rights an duties promoted by the organisation.

In this chapter, we will mainly focus on the Organisation Modelling Language. While there has been a strong emphasis on agent organisations, as shown by the number and diversity of proposed organisational models, a few work aimed at reviewing the proposals and to assess their modelling capabilities [12], at reviewing and comparing organisational paradigms - i.e., general types of organisational structures like hierarchies, teams, markets, matrix organisations, etc. [38, 19], at proposing taxonomies of organisation and social concepts for the engineering of agent organisations [50].

As stated in [15], Multi-Agent organisations exhibit basic traits that participate or not to the modelling proposed by the different approaches cited above. These basic traits that may be part of the organisational models are:

- System structure (resp. functions): elements that form the system and the relationships interconnecting these elements (resp. input/output relations coupling the system to external environment)
- static (resp. kinetic) perspectives: time independent (resp. dependent) description of the system

In the sequel, we will also use the vocabulary introduced in [15]: *organisation models* may give birth to *organisation meta-models*, that is to say a model that represents the conceptualization behind a modelling language. Meta-models are used to produce and define organisation specifications. Organisation specifications are themselves used to implement organisations.

In the following, we will describe different approaches for Organisation Modelling Languages. More specifically, section 1.2 details the $\mathscr{M}$OISE organisation model [41]; whereas section 1.3 details the VOM organisation model [4].

Furthermore, in contemporary complex sociotechnical systems it is not feasible to possess all the information about the environment and to keep this information continuously updated. Agent-oriented modelling as advocated by Sterling & Taveter [59] presents a holistic approach for analysing and designing organisations consisting of humans and technical components. We subsume both under the term of *agent*, which we define as an active entity that can act in the environment, perceive events, and reason [59]. We term organisations consisting of human and man-made agents as *sociotechnical systems*. In section 1.4 we will explain how to apply agent-oriented modelling for describing such agent organisations.

Moreover, in section 1.5, a conceptual metamodel and architecture for Groups in Organized Localities to facilitate the model-based development of agent organisations is briefly explained. Localities capture the idea of a restricted sphere of influence and environmental constraints in which semi-autonomous agents cooperate under the control of centralized regulation bodies, called institutions.

Finally, in section 1.6, a comprehensive view of different Organisation Models is included, in which we compare different organisation models that have been proposed in the literature.

1.2 The $\mathscr{M}$OISE Organisation Model

$\mathscr{M}$OISE (Model of Organisation for multI-agent SystEms) [41] is an organisational model that proposes an Organisation modelling language, an Organisation Management infrastructure and finally basic primitives to make possible the development of Organisation Aware Skills for the agents. We describe below first the $\mathscr{M}$OISE OML.

1.2.1 $\mathscr{M}$OISE Organisation Modelling Language.

The $\mathscr{M}$OISE OML explicitly distinguishes three aspects in the modelling of an organisation: the structural specification, the functional specification and the normative specification.

Structural Specification: The structural specification defines the agents' static relations through the notions of roles, roles relations and groups. A role defines a set of constraints the agent has to accept to enter in a group. There are two kinds of constraints: structural and functional. Structural constraints are defined by means of links and compatibilities that a source role has in relation to a target role. The links are sub-divided in communication, acquaintance and authority links. The communication links enable message exchange between related roles. Acquaintance links enable agents playing one role to get information about agents playing another role. The authority links represent power relation between roles. All the links define constraints that an agent accepts when it enters a group and begins to play a role. By its turn, the compatibility relation constrains the additional roles an agent can play given the roles it is already playing. A compatibility between a role A and a role B means that an agent playing role A is also permitted to play role B. In the structural specification, a group is defined by a group specification. A group specification consists of group roles (roles that can be played), sub-group specifications (group decomposition), links and compatibilities definitions, role cardinalities and sub-group cardinalities.

$\mathscr{M}$OISE Functional Specification: The functional specification describes how an agent organisation usually achieves its global goals, i.e., how these goals are decomposed (by plans) and distributed to the agents (by missions). Global goals, plans and missions are specified by means of a social scheme. A social scheme can be seen as a goal decomposition tree, where the root is a global goal and the leaves are goals that can be achieved by an individual agent. In a social scheme, an internal node and its children represent a plan to achieve a sub-goal. The plan consists in performing the children goals according to a given plan operator. There are three kinds of plan operators: sequence (to do the sub-goal in sequence), choice (to choose and do only one sub-goal) and parallel (to do all the sub-goals in parallel).

$\mathscr{M}$OISE Normative Specification: The normative specification associates roles to missions by means of norms stating permissions and obligations. Norms can also have application-dependent conditions bearing on the organisation or environment state. For instance, norms may define sanction and reward strategies for violation and conformance of other norms. Note that a norm in $\mathscr{M}$OISE is always an obligation or permission to commit to a mission. Goals are therefore indirectly linked to roles since a mission is a set of goals. Prohibitions are assumed 'by default' with respect to the specified missions: if the normative specification does not include a permission or obligation for a role-mission pair, it is assumed that the role does not grant the right to commit to the mission.

1.2.2 $\mathscr{M}$OISE Organisation Model: Other Components

The $\mathscr{M}$OISE organisation model is complemented by an organisation management infrastructure, *ora4mas*, and basic capabilities for making possible the development of organisation aware skills at the agent level.

Organisation Management Infrastructure: The Organisation Management Infrastructure supporting this organisation model follows the Agent & Artifact model [52, 39]. In this approach, a set of organisational artifacts is available in the MAS environment providing operations and observable properties for the agents so that they can interact with the Organisation Management Infrastructure (OMI). For example, each scheme instance is managed by a "scheme artifact". A scheme artifact provides operations such as "commit to mission" and "goal *x* has been achieved" (whereby agents can act upon the scheme) and observable properties (whereby agents can perceive the current state of the scheme). The OMI can be effortlessly distributed by deploying as many artifacts as necessary for the application.

Following the ideas introduced in [40], each organisational artifact has within it an Normative Programming Language interpreter that is given as input: (*i*) the program automatically generated from the organisation specification for the type of the artifact (e.g. the artifact that will manage a social scheme will receive as input the corresponding program translated from that scheme specification), and (*ii*) dynamic facts representing the current state of (part of) the organisation (e.g. the scheme artifact itself will produce dynamic facts related to the current state of the scheme instance). The interpreter is then used to compute: (*i*) whether some operation will bring the organisation into an inconsistent state (where inconsistency is defined by means of the specified regimentations), and (*ii*) the current state of the obligations.

Agent Organisation Aware Mechanisms: Thanks to the *ora4mas* OMI, the set of organisational artifacts, available in the MAS environment, provides operations and observable properties for the agents so that they can interact with the organisation. These different concrete computational entities aimed at managing, outside the agents, the current state of the organisation in terms of groups, social schemes, and normative state encapsulate and enact the organisation behaviour as described by the organisation specifications.

Thanks to the A&A model [52], Artifacts' operations and artifacts' observable properties and events are respectively mapped into agents' external actions and into agents' percepts (leading to beliefs and triggering events). This means that - at runtime - an agent can do an action α if there is (at least) one artifact providing α as operation - if more than one such artifact exist, the agent may contextualise the action explicitly specifying the target artifact. On the perception side, a set of observable properties of the artifacts that an agent is observing are directly represented as (dynamic) beliefs in the agent's belief base - so as soon as their values change, new percepts are generated for the agent that are then processed automatically(within the agent reasoning cycle) and the belief base is updated. So programming an agent, it is possible to write down plans that directly react to changes in the observable state of an artifact or that are selected based on contextual conditions that include the observable state of possibly multiple artifacts. This mapping brings significant improvements to the action and perception model provided in general by agent programming languages.

Translating this to the organisation side, from an agent point of view, organisational artifacts provide the actions that can be used to proactively take part in an organisation (for example, to adopt and leave particular roles, to commit to missions, to signal to the organisation that some social goal has been achieved, etc.), and provide dynamically specific observable properties to make the state of an organisation perceivable along with its evolution. Besides, they provide actions that can be used by organisational agents to manage the organisation itself.

1.3 Modelling Virtual Organisations

The concept of Virtual Organisation (VO) firstly appeared in the business field. *BusinessDictionary.com* defines **Virtual Organisation** as 'an organisation that does not have a physical (bricks and mortar) presence but exists electronically (virtually) on the Internet, or an organisation that is not constrained by the legal definition of a company, or an organisation formed in an informal manner as an alliance of independent legal entities'.

DeSanctis and Monge [17] define a virtual organisation as 'a collection of geographically distributed, functionally and/or culturally diverse entities that are linked by electronic forms of communication and rely on lateral, dynamic relationships for coordination'. Despite its diffuse nature, a common identity holds the organisation together in the minds of members, customers, or other constituents. The virtual organisation is often described as one that is replete with external ties [13], managed via teams that are assembled and disassembled according to needs [33], and consisting of employees who are physically dispersed from one another [11]. The result is a 'company without walls' [27] that acts as a 'collaborative network of people' working together, regardless of location or who 'owns' them [33].

Later the term Virtual Organisation was taken to be used in the research field of computer science. More precisely, in one of the most trending topics in distributed computation, Grid Computing. This field of distributed computation focuses on large-scale, high-performance and innovative systems. Foster *et al.* [26] define a VO as 'a set of individuals and/or institutions defined by sharing computers, software, data, and other resources, as required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering'.

The term Virtual Organisation was also used in Multi-Agent Systems, where this term tries to catch the essence of the concepts from business and grid computing. In this case, the 'Virtual' concept of the Virtual Organisation term normally refers to its 'virtuality', i.e. its software existence. Argente [1] states that a Virtual Organisation is a social entity built by a set of agents that carry out different functionalities. They are structured as a set of communication patterns and a specific topology, following a set of norms, in order to achieve the global goals of the organisation. In fact, this last definition is the one that represents best our idea of Virtual Organisation.

Thus, a Virtual Organisation (VO) presents the following features:

- it is composed by agents, independently from their internal features and individual objectives.
- it follows a global goal, which is not dependant from the agents' individual objectives.
- tasks to be executed by agents are divided by means of roles, which describe the activities and functionalities of the organisation.

- the system is distributed in groups or organisational units where interaction between agents takes place.
- its bounds are clearly defined, determined by the environment of the organisation, the internal and external agents, as well as the functionality and services offered by the organisation.

This section presents two approaches for defining VOs: (i) an UML-based approach, named Virtual Organisation Model (VOM); and a formal approach, named Virtual Organisation Formalisation (VOF).

1.3.1 Virtual Organisation Model (VOM)

The Virtual Organisation Model [4] is an Organisational Modelling Language, defined to describe an Organisation-Centred MAS by means of an UML-based language, identifying the elements that are relevant in an organisation. As most of the metamodels, VOM also gives support to a software development methodology by upholding the development of the Virtual Organisations defined in GORMAS methodology [2] . Systems defined by VOM are structured by means of the Organisational Dimensions [14], which are based on a specific method from the Organisation Theory to define human organisations. Thus, each of these dimensions (structural, functional, dynamical, environment, and normative) is represented by a model inside the Virtual Organisation Model. More specifically, the Organisational Dimensions describe:

- **Structural Dimension**. Describes the components of the system and their relationships. It defines the organisation, composed of agents and organisational units, roles, and their social relationships.
- **Functional Dimension**. Details the functionalities of the system based on services, tasks and objectives. It also describes the stakeholders that interact with the organisational units, the services offered by the organisation, and the resources used by the organisation.
- **Dynamical Dimension**. Defines interactions between agents, as well as the role enactment process, defining the roles that organisational units or agents are able to play.
- **Environment Dimension**. The environment of the organisation is defined by means of the workspaces that structure the environment and the artifacts (that are located inside of the environment). Thus, the organisation can make use of both: workspaces and artifacts.
- **Normative Dimension**. Describes normative restrictions to the action space of entities which populate the system, including organisational norms that agents must fulfil, with associated sanctions and rewards.

As an example, we depict here just a couple of these dimensions, the structural and environment ones, in order to give an overview on how these Organisational

Dimensions are represented by means of VOM. A detailed description of all this model can be found in [4].

The *Structural Dimension* (Fig. 1.1) describes the system's components and their relationships. It allows defining the organisational elements that are independent from the entities that execute them. Specifically, it defines:

- *Organisational Units* (OUs) that build the system, which can also include other units in a recursive way, as well as agents.
- *Roles* defined inside OUs. A role defines the set of functionalities that an entity is able to carry out, and the set of goals and obligations associated to this role. The contains relationship allows to specify the cardinality of each role. A role hierarchy can be defined by means of relationships of inheritance between roles.
- the organisational *social relationships*. The kind of a social relationship between two units is related to their position in the organisational structure (i.e. information, monitoring, and supervision). These relationships allow to describe how roles are related between them, being possible for roles to exchange information, supervise how subordinated roles are developing their objectives, and to delegate their own tasks to subordinated roles.
- *Norms* that control the global behaviour of the members of the organisation.

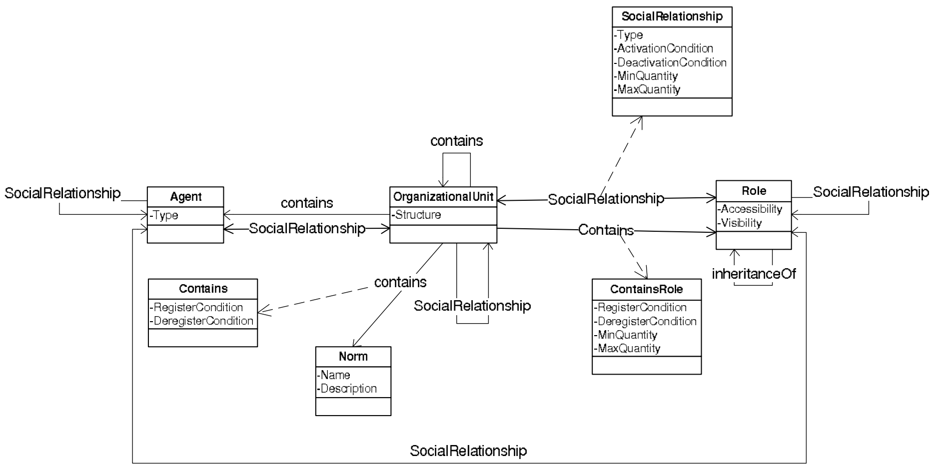


Fig. 1.1 Structural Dimension [4].

The *Environment Dimension* (Figure 1.2) of VOM defines the environment of a Virtual Organisation. It depicts how the environment is structured, adding a physical description, and which are the entities populating it, i.e. the resources that are available for the organisation to be used; or other organisations. This representation of the environment is based on the Agents & Artifacts conceptual framework [52]. The elements on the Environment Dimension are:

- *Workspaces* structure the environment in a similar way than the physical world is structured. They are able to be intersected and nested between them, and organisations are located in one or some of them.
- *Artifacts*, which are reactive entities that agents use to achieve their objectives. Artifacts are located inside workspaces. Each type of artifact is represented in the metamodel by means of its particular operations and observable properties.
- *Agents*, proactive entities of the system (belonging to an organisation or not) that are able to perceive a set of workspaces of the environment and to use a set of artifacts.

The *artifact* entity has been refined into three inherited artifacts, i.e., the Artifacts for Organisational Mechanisms [21], which are a set of artifacts that present features from the Organisational Mechanisms [8]. Organisational Mechanisms enable regulating the behaviour of a MAS in both a macro and a micro perspective. The three types of artifacts defined in VOM are : (i) *Informative artifacts*, provided with operations that allow agents (and other artifacts) to request information; (ii) *incentive artifacts*, whose goal is to modify the reward system of the MAS, and are enhanced with operations for adding and deleting incentives from this reward system; and (iii) *coercive artifacts*, which are able to modify the action space of an agent by means of their particular operations.

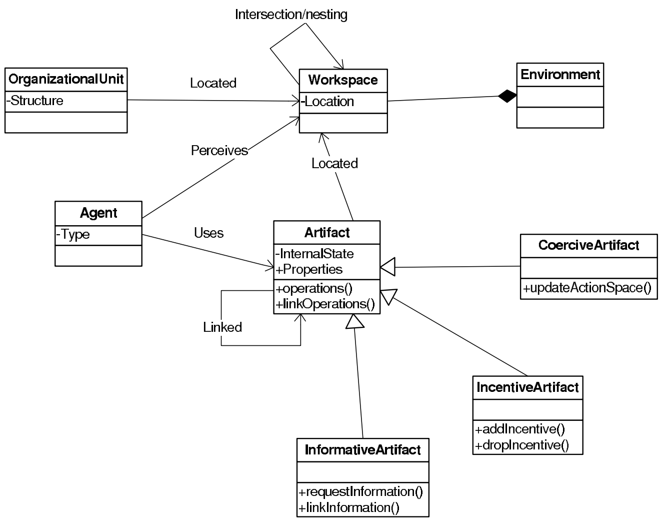


Fig. 1.2 Environment Dimension [4].

1.3.2 Virtual Organisation Formalisation.

This proposal [20] is aimed to cover all concepts of the Organisational Dimensions and to provide a formalization as much complete as possible, with the aim of identifying the elements that compose a VO, facilitating the adaptation process, and checking its correctness.

Virtual Organisation Formalisation (VOF) focuses on three elements: (i) the Organisational Specification (*OS*), which details the set of 'static' elements of the organisation, i.e. the elements that are independent from the final entities that execute them; (ii) the *Organisational Entity* (*OE*), which represents the entities that will then execute the elements in *OS*; and (iii) the *Organisational Dynamics* (ϕ), which relates elements from *OS* with elements from *OE*. As an example, we present here the definition of a VO, and the details for deeper levels of the formalization can be found in [20].

Definition 1 *A Virtual Organisation* $vo \in \mathscr{VO}$ *is defined, at a given time t, as a tuple* $vo(t) = \langle OS(vo,t), OE(vo,t), \phi(vo,t) \rangle$ *where:*

- $OS(vo,t)$ *refers to the* ***Organisational Specification*** *of vo, which describes the structural definition of the organisation, at a given time t. It is defined as* $OS(vo,t) = \langle SD(vo,t), FD(vo,t), ED(vo,t), ND(vo,t) \rangle$ *where:*
  - $SD(vo,t)$ *is the Structural Dimension of vo at a given time t. It defines roles and relations between them.*
  - $FD(vo,t)$ *is the Functional Dimension of vo at a given time t. It describes the functionalities of the system, including goals, services and tasks.*
  - $ED(vo,t)$ *is the Environment Dimension of vo at a given time t, which describes the environment of the organisation, including artifacts and workspaces.*
  - $ND(vo,t)$ *is the Normative Dimension of vo at a given time t, defining the norms that rule a VO.*
- $OE(vo,t)$ *refers to the* ***Organisational Entity*** *of vo at a given time t, which represents the entities populating the system, which can be agents or other VOs.*
- $\phi(vo,t)$ *refers to the* ***Organisational Dynamics*** *of vo at a given time t, allowing to relate* $OS(vo,t)$ *with* $OE(vo,t)$*. It has information about role allocation and active norms and services.*

While VOM is able to define systems at design time, VOF is also able to represent different states that the system passes through its execution. This important feature, as well as its detailed and accurate description of the organisational elements will make it easier to identify different elements that change through time, provoking behaviour or structural changes in the organisation. Thus, VOF will become an excellent tool when dealing with organisational adaptation.

1.4 Agent-Oriented Modelling for Describing Agent Organisations

Agent-oriented modelling as advocated by Sterling & Taveter [59] presents a holistic approach for analysing and designing organisations consisting of humans and technical components. We subsume both under the term of *agent*, which we define as an active entity that can act in the environment, perceive events, and reason [59]. We term organisations consisting of human and man-made agents as *sociotechnical systems*.

The core of agent-oriented modelling lies in the viewpoint framework that can be populated with different kinds of models. Figure 1.3 depicts the viewpoint framework populated with a particular set of models by Sterling & Taveter [59] that we are going to use in Chap. 4 Sect. 4.6 for the case study of designing an adaptive socio-technical system for cell phone manufacturing. The viewpoint framework represented in Figure 1.3 maps each model to the vertical viewpoint aspects of interaction, information, and behaviour and to the horizontal abstraction layers of analysis, design, and platform-specific design. Each cell in the table represents a specific viewpoint. Proceeding by viewpoints, we next give an overview of the types of models employed in Chap. 4 Sect. 4.6.

	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behaviour
Analysis	Role models and organization model	Domain model	Goal models and motivational scenarios
Design	Agent models, acquaintance model, and interaction models	Knowledge model	Scenarios and behaviour models
Platform-specific design	Platform-specific design models		

Fig. 1.3 The model types of agent-oriented modelling

From the viewpoint of *behaviour analysis*, a *goal model* can be considered as a container of three components: goals, quality goals, and roles [59]. A *goal* is a representation of a functional requirement of the sociotechnical system to be developed. A *quality goal*, as its name implies, is a non-functional or quality requirement of the system. Goals and quality goals can be further decomposed into smaller related subgoals and subquality goals. The hierarchical structure is to show that the subcomponent is an aspect of the top-level component. Goal models also determine roles that are capacities or positions that agents playing the roles need to contribute to achieving the goals. Roles are modelled in detail in the viewpoint of interaction analysis. The notation for representing goals and roles is shown in Figure 1.4. This notation is used in Chap. 4 Sect. 4.6 in presenting requirements for the case study of an adaptive socio-technical system for cell phone manufacturing. Goal models go hand in hand with *motivational scenarios* that describe in an informal and loose narrative manner how goals are to be achieved by agents enacting the corresponding roles [59].

From the viewpoint of *interaction analysis*, the properties of roles are expressed by role models. A *role model* describes the role in terms of the responsibilities and constraints pertaining to the agent(s) playing the role. *Organisation model* is a model that represents the relationships between the roles of the sociotechnical system, forming an organisation [59]. Organisation models are central in designing sociotechnical systems because organisational relationships between roles essentially determine interaction between roles in an organisation. Interactions will be addressed from the viewpoint of interaction design.

From the viewpoint of *information analysis*, *domain model* represents the knowledge to be handled by the sociotechnical system. A domain model consists of domain entities and relationships between them. A domain entity is a modular unit of knowledge handled by a sociotechnical system [59].

From the viewpoint of *interaction design*, *agent models* transform the abstract constructs from the analysis stage, roles, to design constructs, *agent types*, which will be realized in the implementation process. The *acquaintance model* complements the agent models by outlining interaction pathways between the agents of the system. *Interaction models* represent interaction patterns between agents of the given types. They are based on responsibilities defined for the corresponding roles.

From the viewpoint of *information design*, the *knowledge model* describes the private and shared knowledge by agents of the Multi-Agent System to be designed. Finally, from the perspective of *behaviour design*, *scenarios* and *behaviour models* describe the behaviours of agents in the system.

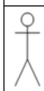
Symbol	**Meaning**
	Goal
	Quality goal
	Role
	Relationship between goals
	Relationship between goals and quality goals

Fig. 1.4 Notation for modelling goals and roles

We described one possible way of populating the viewpoint framework with models. Agent-oriented modelling is a generic approach rather than another AOSE methodology. It means that rather than using particular types of models, the completeness of the design process matters. Design is complete when all the viewpoints corresponding to the cells of Table 1.3 are covered by models. For example, in Chap. 7 of [59] it is demonstrated how the viewpoint framework can be populated by (combinations of) models originating in the following AOSE methodologies: Gaia [9], MaSE [16], Tropos [7], Prometheus [53], ROADMAP [45], and RAP/AOR [61]. Agent-oriented modelling thus prescribes neither any specific agent-oriented software engineering methodology nor any agent-based software platform, but is compatible with most of them. Agent-oriented modelling instead proposes a conceptual framework that facilitates achieving the completeness of views and abstraction layers when designing a sociotechnical system, such as an information system or industrial automation system.

In Chap. 4 Sect. 4.6 we will show how agent-oriented modelling can be applied to designing adaptive agent organisations. Our starting point that Section 4.6 is that adaptivity needs to be part of overall system design [58].

1.5 Decribing Agent Organisations with Groups of Autonomous Agents in Organized Localities

Agent organisation systems are characterized by loosely coupled, software-controlled systems that cooperate to achieve joint goals. Each system operates semi-autonomously in order to pursue individual tasks, but it also obeys the current constraints within its local environment.

The assumption is that subsystems are developed independently due to their purposes and unifying requirements of an entire system. New control challenges arise from a shift from traditional hierarchical organisation to a Multi-Agent Systems organisation. But it also opens ample of new opportunities in terms of ad-hoc coordination and co-operation in order to maximize throughput and avoid breakdowns of agent organisations.

The integration of subsystems and the growing complexity of joint tasks, the need for ”semantically rich abstract levels of description” [36] increases, specially social concepts like organisations, institutions and norms [49, 62]. Social concepts are a means of explicit representation of global objectives and constraints and of their relation to the level of interacting groups and even to individuals with their beliefs, desires, and intentions (BDI).

A conceptual metamodel and architecture is described for Groups in Organized Localities to facilitate the model-based development of agent organisations. Localities capture the idea of a restricted sphere of influence [44] and environmental constraints in which semi-autonomous agents cooperate under the control of centralized regulation bodies, called institutions.

Agent Organisations. They can be represented by the integration of four dimensions, introduced by Huhn et al. [43] which consists of the interacting loop:

1. The *Environment* is represented as the *Locality* which is scanned by the agent and he performs action inside.
2. The *Agent* has an architecture with an *Execution Layer* where the agent is connected to the *Locality* and *LocalityRole* is allocated to the agent.
3. The *Organisation* is the connection between the *Agent* and the *MAS* where it is embedded together with *Institution*.
4. The *Institution* gives (structural, functional or deontic) rules and norms to the sphere of influence to the so-called *Locality*.

Representation of the Environment. To handle the environment's complexity the focus is just on the significant parts and to extract, collect, and pre-process important information about its state. Besides this filtering process the division of the global environment into smaller, well-defined local sections with specific properties and constraints, called **organized localities** is necessary. The locality is decomposed into several scenes and each scene is characterized by constraints which may take effect on different levels of the system.

An organized locality can be understood as a physical or virtual place offering a number of opportunities. It has a scope defining a boundary, so systems may enter, leave, and return later to the locality. Further a locality may provide organisations to foster coordination. It is associated with the concept of institutions to regulate the interaction of autonomous, heterogeneous agents beyond physical and technical constraints. They regulate the agent behaviour in order to balance between different interests and to establish and sustain certain notions of stability. Organisations structure the grouping and collaboration of agents within the locality.

In order to provide the structure of the localities, we need a representation of the environment, which enables a proper association between the specific regulation mechanisms and the localities. The institutions which are associated with a locality, provide regulation mechanisms within the scope of a specific scene. The division into scenes can be motivated by various tasks rules, processes, requirements, properties, constraints or resources (e.g. sensor properties, movement constrains or energy resources). Within these scenes, associated sets of norms are used to regulate the behaviour and interaction of the agents. According to this, the agents need an internal representation of the context which is relevant in the specific scope. The locality is defined as a virtual infrastructure to be used by the agents to achieve goals related to the subject of the locality. An approach towards adaptive IT-ecosystems is given in Rausch et al. [55] and especially how to create an environment standard is specified in Behrens et al. [5, 6]. Practical approaches are done in Görmer et al. [30] for integrating also institutions, Chu et al. [10] for combining tools for agent-based traffic behaviour which the novel traffic context for adaptive systems is described in Görmer et al. [29] and an application in Görmer & Mumme [32] for cooperative traffic behaviour.

Representation of the Agents. Based on the design of intelligent agents of Müller [51], also Huhn et al. [43] propose an *agent architecture* with four layers: *Social Context Layer* (SCL), *Individual Context Layer* (ICL), *Execution Layer* (EL), and *Mechatronic Layer* (ML) (see Fig. 1.5). Agents perform predefined atomic or sequenced (plans) actions related to their goals. Goals and plans are potentially spread among multiple agents (joint goals/plans). Each layer has an authority. If

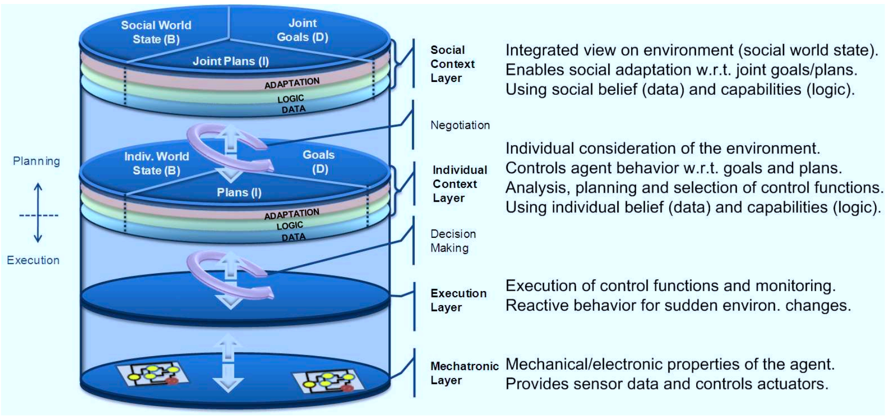


Fig. 1.5 AAOL Agent Architecture [43]

multiple agents act in the same locality, joint tasks have to be coordinated in groups and resource conflicts need to be solved.

Relations of an agent in a metamodel are described according to Fischer [35]: an agent has accesses to a set of resources (information, knowledge, ontologies, etc.) from its environment, i.e., the locality. Furthermore, an agent has goals and is able to take on locality roles (to act in accordance to a plan) and behaviours, which are represented by the agents' capabilities. By acting the agent receives positive or negative rewards. Additionally Fischer uses the concept of *Instances* that can be considered as run-time objects of an agent that defines the corresponding type.

Representation of the Organisation. In the agent architecture described in Fig. 1.5 organisations are located in the *Social Context Layer* (SCL) and can be seen as computational methods inspired by concepts from economy and sociology that appear as one entity in the locality based upon social and functional distinctions and roles amongst individuals. Organisations can also be structured hierarchically e.g. by providing certain agents with more authority than others through role definitions. A peer-to-peer architecture is any distributed network composed of agents that make a portion of their resources directly available to other agents, without the need for central coordination instances. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model. The fully connected architecture has a general form of a chief director usually forming the single well-informed element, the so-called ”voice” of an organisation to the outside, to sub-division managers and to the workers. A group can be seen as a specialized entity (or subsystem) of an organisation, usually consisting only of one leader and workers to reach a common goal or achieve a joint plan. For this, communication, negotiation and conflict resolution is connecting the individual with the social context layer. The connections between the agents with different roles imply interaction guaranteeing the service of the localities; this may lead to conflicts between agents which need to be handled like in Le et al. [47].

Fig. 1.6 is an extension of [35] and shows the metamodel of organisations. It includes the concept of an *Organisation* and its *Structure*, *Group* and its *Context*, *Institution* and *Norm*, *Binding*, *InteractionUse*, *ActorBinding*, *Interaction* and its *Protocols* for *Communication* and *Coordination*, *LocalityRole*, *Actor* and *Agent* as well as *Capability* and *Resource* (from the agent aspect). An organisation is derived from the agent perspective and it inherits characteristics of an agent [35], i.e. capabilities which can be performed by its members. A *Group* is a special kind of an organisation that is bound by a *Group context*. The *Structure* defines the pattern of the organisation. It can bind agents or organisations to the *LocalityRole*. *Interaction* in an organisation has internal protocols that specify how its members communicate with each other and coordinate their activities. For interaction, *LocalityRole*s are bound to *Actor*s (by *ActorBinding*) that can be considered as representative entities within the corresponding interaction protocols. Thus, an actor can be seen as an agent (or organisation) with a *Role* and a task.

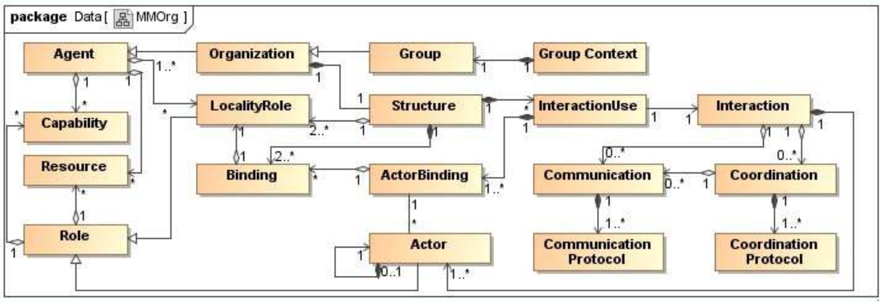


Fig. 1.6 Metamodel of Organisations and Roles [43]

A role defines the behaviour of an agent in a given context (e.g., an organisation). Therefore it provides an agent with capabilities and a set of resources it has access to. An actor can be considered as a generic concept and either binds instances directly or through the concepts LocalityRole and Binding. The set of bound entities could be further specialized through the subactor (specialization of the superactor) reference that refers again to an actor.

Grouping allows an agent to extend its range of perception (RoP) by exchanging information with other members. Agents are coordinated at group level. Group-oriented coordination allows agents e.g. to form faster and slower agent groups like in Görmer & Müller [31]. In Chap. 4 Sect. 4.8 a detailed description of group-oriented coordination is given.

Representation of the Institution. Institution is associated with a locality and provides normative regulations (norms) and mechanisms to establish or to ensure their compliance. It acts through an organisation that executes institutional tasks. The tasks contributing to norm compliance are:

1. An *information service* administers the identities of agents currently present in the locality and provides them with knowledge about the current norms,
2. *Norm monitors* monitor whether the agents behave according to the norms based on the information gathered from *observers*,
3. A *norm enforcement* guarantees that control is imposed on the agents participating in the locality in such a way that they will behave norm-compliant to assure vital global objectives and the safety of individuals.

Norms are an explicit description of the regulations that govern the agents' behaviour in the locality for the benefit of the community and itself as a member of it. In the approach of Huhn et al. [43], norms are defined by the institution in a top-down manner and they consider that the agents are able to understand these norms.

Huhn et al. uses institutional agents (IAs), which act preemptively on agents only in case of obligations. At each step, the IAs compute a list of candidates of agents, for which an obligation applies. For each candidate, the IAs then identify forbidden actions, from the list of possible actions defined at design time. Only at this moment the IA acts and restricts the candidates from performing the forbidden actions. The other types of norms are handled by means of rewards and sanctions. A more detailed study is described in Klar & Huhn [46] for interfaces and models.

A main benefit of the described approach based on Huhn et al. is that its concepts (localities, institutions, and norms) provide designers with instruments for flexible modelling of different control topologies of agent organisations, ranging from centralized and homogeneous to decentralized and heterogeneous settings. Further, the multi-agent based approach in conjunction with the localities concept supports well decentralised systems design scenarios, where the different parts evolve independently from each other while having to obey certain invariants or rules constraining the overall structural or behavioural development of agent organisations.

1.6 Conclusion and Discussion

In this chapter we have detailed four organisational models. Considering the synthesis presented in [15], where the authors have analyzed existing organisation models (MOISE, AGR [24], TAEMS [15, 48], ISLANDER [22], OperA [18], AGRE [25], MOISEInst [28], ODML [37], STEAM [60], AUML [54], MAS-ML [57]), different modelling dimensions have been exhibited (cf. Table 1.1).

It is shown that an organisational model may provide constructs to represent formal patterns in the structure and functions of an agent organisation, these patterns being either static or kinetic. This general analysis lead to posit four cohesive categories of modelling constructs in an organisational model:

- *Organisational Structure:* constructs to represent what aspects of the structure of the agent organisation have to be invariant through time;
- *Organisational Functions:* constructs that represent global goals and goal decompositions to be accomplished by the agent organisation;
- *Organisational Interactions:* constructs to represent time-dependent aspects of standardized actions and interactions involving the elements from the organisational structure and organisation function;
- *Organisational Norms:* constructs to further regulate and show how organisational structure (time-independent relations), organisational interaction (time-dependent functioning) and organisational functions are interrelated.

Beyond these dimensions that are mostly found in existing organisational models, other complementary traits of agent organisations have been found:

- *Organisational Environment:* constructs to represent a collection of resources in the space of the agent organisation formed by non-autonomous entities that can be perceived and acted upon (manipulated, consumed, produced, etc.) by the components agents;
- *Organisational Evolution:* constructs to model changes in the organisation (formal structure, norms and goals) at some points in the time in order to adapt the functioning of the agent organisation to new demands from the environment;
- *Organisational Evaluation:* constructs to measure the performance of the formal structure and norms of an agent organisation w.r.t. specific goals;
- *Organisational Ontologies:* constructs to build conceptualizations regarding the application domain of the agent organisation that must be consistently shared by the component agents. These global conceptualizations are important to maintain the coherence of the activity inside the agent organisation.

The models detailed in this chapter (VOM, Agent-Oriented Modelling and Autonomous Agents in Organized Localities (AAOL)), confirm the existence of these dimensions and the diversity of constructs proposed in the Multi-Agent literature to define organisation for agents to coordinate in decentralized and open systems. More specifically:

- Organisational Structure - in almost all models this is the primary modelling concern. The main modelling elements found were roles, groups, and relationships between them. The structure of roles and groups defines a system of possible positions where the agents should find a place to become a member of an agent organisation.
- Organisational Interactions - found mainly in ISLANDER and OperA. In this respect, the models provide constructs to express the dynamic of communicative interactions between the agents (positioned in the social structure). Some constructs are interaction protocols, scenes and scene structures. In AAOL it is found in Görmer et al. [30] to evaluate the system with an interaction level to combine the micro and macro level of a Multi-Agent System.

Model	Structure	Interaction	Function	Norms	Environment	Evolution	Evaluation	Ontology
AGR	+	+	-	-	-	-	-	-
TAEMS	-	-	+	-	+	-	+	-
ISLANDER	+	+	-	+	-	-	-	+
OperA	+	+	+	+	-	-	-	+
AGRE	+	+	-	-	+	-	-	-
MOISEInst	+	-	+	+	-	+	-	-
ODML	+	-	-	-	-	-	+	-
STEAM	+	-	+	-	-	-	-	-
AUML	+	+	+	-	+	-	-	-
MAS-ML	+	+	+	+	+	-	-	-
$\mathcal{M}$OISE	+	-	+	+	-	+	-	-
VOM	+	+	+	+	+	-	-	+
Agent-Oriented	+	+	+	+-	+	-	-	-
AAOL	+	+-	+-	+	+	+	+-	-

Table 1.1 Organisation Modelling Dimension in some organisational models.

- Organisational Function - appeared with more emphasis in TAEMS, STEAM and MOISE+. In these models, (one of) the main concern is to provide means to specify procedures to achieve goals. In order to model this feature, we find in the models conceptual elements such as tasks or goals, missions and plans. In AAOL it is designed in its structure on the individual and global context layer.
- Organisational Norms - described in term of deontic norms (regulate the behaviour of social entities: what they are allowed to do - direct or indirectly -, what they are obliged to do, etc.). ISLANDER, OperA, MOISEInst and AAOL are representative examples of organisational models that provide mechanisms to specify normative structures.
- Organisational Environment - here the models provide means to describe elements lying in the topological space occupied by the agent organisation and the way agents (positioned in the social structure, performing some task and/or in the course of some dialogical interaction, respecting some norms) are related to these elements. AGRE, MAS-ML and AAOL are examples of organisational models (modelling techniques) that provide constructs to represent organisation environment elements. MOISE+ and VOM define environment by means of the Agents and Artifacts (A&A) conceptual framework.
- Organisational Evolution - this is related to modelling the way organisations can change (their social, task decomposition, dialogical, and normative structures) in order to cope with changes in its purpose and/or environment. Among the organisational models reviewed, MOISE+ and its extension MOISEInst explicit address organisation evolution issues. AAOL has a big focus on adaptivity and controlling in order to achieve a system balance of an IT-ecosystem.
- Organisational Evaluation - in order to modify some organisation (re-organisation) it is important to know how well the present organisation is performing. Thus, some models have elements to specify means to assess some properties of an or-

ganisation. Among these we have found TAEMS and ODML. Partial evaluation is also done by AAOL.

- Organisational Ontology - here we find ontologies used to ground the elements of the other dimensions as can be seen in the organisational models ISLANDER and OperA, and to define mental states of the agents in VOM. In AAOL there exists also works for ontologies.

References

1. Argente, E.: GORMAS: Guias para el desarrollo de Sistemas multi-agente abiertos basados en organizaciones. Ph.D. thesis, Universidad Politecnica de Valencia (2008)
2. Argente, E., Botti, V., Julian, V.: GORMAS: An Organizational-Oriented Methodological Guideline for Open MAS. In: Proc. Agent-Oriented Software Engineering. pp. 85–96 (2009)
3. Argente, E., Botti, V., Carrascosa, C., Giret, A., Julián, V., Rebollo, M.: An abstract architecture for virtual organizations: The THOMAS approach. Knowl. Inf. Syst. 29(2), 379–403 (2011)
4. Argente, E., Julian, V., Botti, V.: MAS Modeling based on Organizations. In: PostProceedings 9th International Workshop AOSE'08. vol. 5386, pp. 16–30. Springer (2009)
5. Behrens, T.M., Hindriks, K.V., Dix, J.: Towards an environment interface standard for agent platforms. Ann. Math. Artif. Intell. 61(4), 261–295 (2011)
6. Behrens, T.M., Hindriks, K.V., Bordini, R., Braubach, L., Dastani, M., Dix, J., Hübner, J., Pokahr, A.: An interface for agent-environment interaction. In: Collier, R., Dix, J., Novák, P. (eds.) Programming Multi-Agent Systems, Lecture Notes in Computer Science, vol. 6599, pp. 139–158. Springer Berlin / Heidelberg (2012)
7. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multiagent Systems 8(3), 203–236 (2004)
8. Centeno, R., Billhardt, H., Hermoso, R., Ossowski, S.: Organising MAS: a formal model based on organisational mechanisms. In: Proceedings of the 2009 ACM symposium on Applied Computing. pp. 740–746. ACM (2009)
9. Cernuzzi, L., Juan, T., Sterling, L., Zambonelli, F.: The GAIA methodology: Basic concepts and extensions, pp. 69–88. Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook, Kluwer Publishing (2004)
10. Chu, V.H., Görmer, J., Müller, J.P.: Atsim: Combining aimsum and jade for agent-based traffic simulation. In: Avances en Inteligencia Artificial. Actas de CAEPIA '11. vol. 1 (2011)
11. Clancy, T.: The latest word from thoughtful executives. Academy of Management Executive 8(5), 8–10 (1994)
12. Coutinho, L.R., Sichman, J.S., Boissier, O.: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, chap. Modelling Dimensions for Agent Organizations, pp. 18–50. Information Science Reference Publisher (2009)
13. Coyle, J., Coyle, M., Schnarr, N.: The Soft-Side Challenges of the Virtual Corporation. Human Resource Planning 18, 41–42 (1995)
14. Criado, N., Argente, E., Julian, V., Botti, V.: Designing virtual organizations. In: Proc. International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 440–449 (2009)
15. Decker, K.S.: TÆMS: A framework for environment centered analysis and design of coordination mechanisms. In: O'Hare, G.M.P., Jennings, N.R. (eds.) Fundations of Distributed Artificial Intelligence, chap. 16, pp. 429–447. John Wiley, New York (1996)
16. DeLoach, S.A., Kumar, M.: Multiagent systems engineering: An overview and case study, pp. 317–340. Agent-Oriented Methodologies, Idea Group (2005)

17. DeSanctis, G., Monge, P.: Communication processes for virtual organizations. Journal of Computer-Mediated Communication 3(4), 0–0 (1998)
18. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis, Universiteit Utrecht (2004)
19. Dignum, V., Dignum, F.: Progress in Artificial Intelligence, 10th Portuguese Int. Conf. on Artificial Intelligence, EPIA'01, vol. LNAI, chap. Modelling agent societies: co-ordination frameworks and institutions, pp. 191–204. Springer Verlag (2001)
20. Esparcia, S., Argente, E.: Formalizing Virtual Organizations. In: 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011). vol. 2, pp. 84–93. SciTePress (2011)
21. Esparcia, S., Argente, E., Centeno, R., Hermoso, R.: Enhancing MAS Environments with Organizational Mechanisms. In: Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on. vol. 1, pp. 457–464. IEEE (2010)
22. Esteva, M., Padget, J.A., Sierra, C.: Formalizing a language for institutions and norms. In: Meyer, J.J.C., Tambe, M. (eds.) ATAL. Lecture Notes in Computer Science, vol. 2333, pp. 348–366. Springer (2001)
23. Esteva, M., Rodríguez-Aguilar, J.A., Rosell, B., L., J.: AMELI: An agent-based middleware for electronic institutions. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). pp. 236–243. ACM, New York (2004)
24. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J. (eds.) AOSE. Lecture Notes in Computer Science, vol. 2935, pp. 214–230. Springer (2003)
25. Ferber, J., Michel, F., Baez, J.: AGRE: Integrating environments with organizations. Environments for Multi-agent Systems. pp. 48–59. Springer (2005)
26. Foster, I., Kesselman, C.: The Anatomy of the Grid - Enabling Scalable Virtual Organizations. Int. J. Supercomput. Appl. 15, 200–222 (2001)
27. Galbraith, J.: Designing organizations. Jossey-bass (1995)
28. Gâteau, B., Boissier, O., Khadraoui, D., Dubois, E.: Moiseinst: An organizational model for specifying rights and duties of autonomous agents. In: Third European workshop on multi-agent systems (EUMAS 2005). pp. 484–485 (2005)
29. Görmer, J., Ehmke, J.F., Fiosins, M., Schmidt, D., Schumacher, H., Tchouankem, H.: Decision support for dynamic city traffic management using vehicular communication. In: SIMULTECH. pp. 327–332 (2011)
30. Görmer, J., Homoceanu, G., Mumme, C., Huhn, M., Müller, J.P.: Jrep: Extending repast simphony for jade agent behavior components. In: IAT. pp. 149–154 (2011)
31. Görmer, J., Müller, J.P.: Multiagent system architecture and method for group-oriented traffic coordination. In: Proceedings of IEEE DEST 2012. IEEE press (2012)
32. Görmer, J., Mumme, C.: Multiagentensysteme für das kooperative verkehrsmanagement. In: Back, A., Bick, M., Breunig, M., Pousttchi, K., Thiesse, F. (eds.) MMS 2012: Mobile und ubiquitäre Informationssysteme. Proceedings der 7. Tagung, 1. bis 2. März 2012 in Braunschweig, Deutschland. pp. 138–142 (2012)
33. Grenier, R., Metes, G.: Going virtual: Moving your organization into the 21st century. Prentice Hall PTR (1995)
34. Gutknecht, O., Ferber, J.: The madkit agent platform architecture. In: Revised Papers from the Int. Workshop on Infrastructure for Multi-Agent Systems: Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. pp. 48–55. Springer-Verlag, London, UK (2001)
35. Hahn, C., Mora, C.M., Fischer, K.: A platform-independent metamodel for Multiagent Systems. Autonomous Agents and Multi-Agent Systems 18(2), 239–266 (2009)
36. Heistracher, T., Kurz, T., Masuch, C., Ferronata, P., Vidal, M., Corallo, A., Briscoe, G., Dini, P.: Pervasive service architecture for a Digital Ecosystem. In: 1st WS on Coordination and Adaptation Techniques for Software Entities (2004)
37. Horling, B., Lesser, V.: Quantitative organizational models for large-scale agent systems. In: Ishida, T., Gasser, L., Nakashima, H. (eds.) MMAS. Lecture Notes in Computer Science, vol. 3446, pp. 121–135. Springer (2004)

38. Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms. The Knowledge Engineering Review 19(04), 281–316 (Nov 2005)
39. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents. Journal of Autonomous Agents and Multi-Agent Systems (2009)
40. Hübner, J.F., Boissier, O., Bordini, R.H.: From organisation specification to normative programming in multi-agent organisations. In: CLIMA XI. pp. 117–134 (2010)
41. Hübner, J.F., Sichman, J., Boissier, O.: A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: Bittencourt, G., Ramalho, G. (eds.) Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2507, pp. 439–448. Springer Berlin / Heidelberg (2002)
42. Hübner, J.F., Sichman, J.S., Boissier, O.: $\mathscr{S}$-$\mathscr{M}$OISE$^+$: A middleware for developing organised multi-agent systems. In: Boissier, O., Dignum, V., Matson, E., Sichman, J.S. (eds.) Proceedings of the International Workshop on Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS (OOOP'2005). LNCS, vol. 3913. Springer (2006)
43. Huhn, M., Müller, J., Görmer, J., Homoceanu, G., Le, N.T., Märtin, L., Mumme, C., Schulz, C., Pinkwart, N., Müller-Schloer, C.: Autonomous agents in organized localities regulated by institutions. In: 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST). pp. 54–61 (2011)
44. Jennings, N.R.: On agent-based Software Engineering. Artificial Intelligence 177(2), 277–296 (2000)
45. Juan, T., Pearce, A., Sterling, L.: Roadmap: Extending the gaia methodology for complex open systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002). pp. 3–10. ACM, Bologna, Italy (2002)
46. Klar, D., Huhn, M.: Interfaces and Models for the Diagnosis of Cyber-Physical Ecosystems. In: Proceedings of IEEE DEST 2012. IEEE press (2012)
47. Le, N.T., Märtin, L., Mumme, C., Pinkwart, N.: Communication-free detection of resource conflicts in multi-agent-based cyber-physical systems. In: Proceedings of IEEE DEST 2012. IEEE press (2012)
48. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P., Zhang, X.: Evolution of the gpgp/taems domain-independent coordination framework. Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers 9(1), 87–143 (July 2004)
49. Lopes Cardoso, H., Oliveira, E.: Electronic Institutions for B2B: Dynamic normative environments. Artificial Intelligence and Law 16(1), 107–128 (2008)
50. Mao, X., Yu, E.: Organizational and social concepts in agent oriented software engineering. In: Odell, J., Giorgini, P., Müller, J.P. (eds.) AOSE. Lecture Notes in Computer Science, vol. 3382, pp. 1–15. Springer (2004)
51. Müller, J.P.: The Design of Intelligent Agents, Lecture Notes in Artificial Intelligence, vol. 1177. Springer-Verlag (1996)
52. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (Dec 2008)
53. Padgham, L., Winikoff, M.: Developing intelligent agent systems: A practical guide. John Wiley and Sons (2004)
54. Parunak, H.V.D., Odell, J.: Representing social structures in UML. In: Wooldridge, M., Weiß, G., Ciancarini, P. (eds.) AOSE. Lecture Notes in Computer Science, vol. 2222, pp. 1–16. Springer (2001)
55. Rausch, A., Müller, J., Goltz, U., Niebuhr, D.: It ecosystems: A new paradigm for engineering complex adaptive software systems. In: Proceedings of IEEE DEST 2012. IEEE press (2012)
56. Scott, W.R.: Organizations: Rational, Natural, and Open Systems. Prentice Hall, Englewood Cliffs, New Jersey (1981)
57. da Silva, V.T., Choren, R., de Lucena, C.J.P.: A UML based approach for modeling and implementing multi-agent systems. Autonomous Agents and Multiagent Systems, International Joint Conference on 2, 914–921 (2004)

58. Sterling, L.: Adaptive: A quality goal for agent-oriented models or a fundamental feature of agents? In: Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC). Milano, Italy (2011)
59. Sterling, L., Taveter, K.: The Art of Agent-Oriented Modeling. The MIT Press, Cambridge, MA, and London, England (2009)
60. Tambe, M., Adibi, J., Alonaizon, Y., Erdem, A., Kaminka, G.A., Marsella, S., Muslea, I.: Building agent teams using an explicit teamwork model and learning. Artif. Intell. 110(2), 215–239 (1999)
61. Taveter, K., Wagner, G.: Towards radical agent-oriented software engineering processes based on AOR modelling, pp. 277–316. Agent-Oriented Methodologies, Idea Group (2005)
62. Vázquez-Salceda, J., Dignum, V., Dignum, F.: Organizing multiagent systems. Autonomous Agents and Multi-Agent Systems 11, 307–360 (November 2005)

Chapter 2
Modelling Agent Institutions

Nicoletta Fornara and Henrique Lopes Cardoso and Pablo Noriega and Eugenio Oliveira and Charalampos Tampitsikas

Abstract Everyday uses of the notion of institution and some typical institutions have been studied and formalized by economists and philosophers. Borrowing from these everyday understandings, and influenced by their formalizations, the notion of institution has been used within the agents community to model and implement a variety of socio-technical systems. Their main purpose is to *enable* and *regulate* the interaction among autonomous agents in order to achieve some collective endeavour. In this chapter we present and compare three frameworks for agent-based institutions (i) ANTE, a model that considers electronic institutions as computational realizations of adaptive artificial environments for governing multi-agent interactions; (ii) OCeAN, extended in MANET, a model for specifying Artificial Institutions (AIs), situated in agent environments, which can be used in the design and

Nicoletta Fornara
Università della Svizzera italiana, via G. Buffi 13, 6900 Lugano, Switzerland e-mail: `nicoletta.fornara@usi.ch`. Supported by the Hasler Foundation project nr. 11115-KG.

Henrique Lopes Cardoso
LIACC / Dep. Eng. Informática, Faculdade de Engenharia, Universidade do Porto Rua Dr. Roberto Frias, 4200-465 Porto, Portugal e-mail: `hlc@fe.up.pt`

Pablo Noriega
IIIA, Artificial Intelligence Research Institute, CSIC, Spanish Scientific Research Council e-mail: `pablo@iiia.csic.es`. Partially supported by the Consolider AT project CSD2007-0022 INGENIO 2010 of the Spanish Ministry of Science and Innovation and Generalitat de Catalunya 2009-SGR-1434

Eugenio Oliveira
LIACC / Dep. Eng. Informática, Faculdade de Engenharia, Universidade do Porto Rua Dr. Roberto Frias, 4200-465 Porto, Portugal e-mail: `eco@fe.up.pt`

Charalampos Tampitsikas
Università della Svizzera italiana, via G. Buffi 13, 6900 Lugano, Switzerland, University of Applied Sciences Western Switzerland, 3960 Sierre, Switzerland e-mail: `charalampos.tampitsikas@usi.ch`. Supported by the Swiss State Secretariat for Education and Research, COST Action IC0801 “Agreement Technologies”, project title “Open Interaction Frameworks, towards a Governing Environment”.

implementation of different open interaction systems; and (iii) a conceptual core model for Electronic Institutions (EIs), extended with EIDE, based on open, social, decomposable and dialogical interactions. Open challenges in the specifications and use of institutions for the realization of real open interaction systems are discussed.

2.1 Introduction

In everyday language, the notion of "institution" is used in different contexts, for example when one talks about the "institution of marriage", when we say that a given university is an "institution of higher education", or when we say that a politician does not behave "institutionally". Those everyday uses and some typical institutions have been studied and formalized by economists, political scientists, legal theorists and philosophers (see [2, 49]). There are three features that these conventional understandings have. The first is the distinction between "institutional" and "brute" (or actual, physical or real) facts [51, 34], and the correspondence between the two. Another key conceptual element is the separation between the institution itself and the agents that participate in the collective endeavor that is the purpose of the institution. Finally, the assumption that institutions involve regulations, norms, conventions and therefore some mechanism of governance that make those components effective. In fact, most theoretical approaches to conventional institutions may be distinguished by the way this last assumption is made operational. In particular, while some approaches (for instance North [44] and Ostrom [46]) take institutions to be the conventions themselves–and consequently draw a clear distinction between institutions (conventions) and organisations (the entities that put the conventions in practice)—others (like Simon [52]) take institutions to be organisations (with rules or norms, institutional objects and due processes or procedures) but still keep individuals out of the institution.

Borrowing from these everyday understandings, and influenced by their formalizations, the notion of institution has been used within the agents community to model and implement a variety of socio-technical systems that serve the same purposes that conventional institutions serve. Artificial, electronic, agent-mediated, agent-based or, simply, agent institutions are some of the terms that have been used to name such computational incarnations of conventional institutions in the agents community, and for the sake of economy we take them as synonymous in this introduction. Their main purpose is to *enable* and *regulate* the interaction among autonomous agents in order to achieve some collective endeavour.

These agent institutions, as agent-based organisations do, play a crucial role as *agreement technologies* because they allow to specify, implement and enact the conventions and the services that enable the establishment, execution, monitoring and enforcement of agreements among interacting agents.

Agent institutions have been implemented as multi-agent systems using different "frameworks" (conceptual models that have associated tools and a software architecture that allow implementation of particular institutions). However, these artificial institutions all hold three assumptions that mirror the three features of conventional institutions mentioned above:

1. Institution, on one hand, and agents, on the other, are taken as first-class entities. A particular institution is specified through a conceptual model, based on a metamodel, that may be more or less formalized, then it may be implemented on some type of institutional environment and enacted through interactions of some participating entities.
2. Institutions are open MAS, in the sense that: (i) it is not known in advance what agents may participate in an enactment, now when these agents may decide to enter or leave an enactment; (ii) the institution does not know what the particular goals of individual agents are; (iii) the institution has no control over the internal decision-making of agents (iv) agents may not necessarily comply with institutional conventions.
3. Institutions are regulated systems. Interactions in the agent institution must comply with some conventions, rules, and norms that apply to every participant agent and are somehow enforced. Regulations control interactions and are applicable to individual agents in virtue of the activities they perform and not because of who they are.

There are several ways that these assumptions lead to more precise notions of what constitutes an institution and how these may be implemented. This chapter discusses three frameworks that actually achieve that objective but before discussing those frameworks we would like to provide some background.

Institutions are Normative MAS. Institutions are a class of "normative multi-agent systems" (norMAS) [7, 6]:

> A *normative multi-agent system* is a multi-agent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fullfilment.

The ground assumption in normative MAS is that norms are used to constrain undesired behaviour, on one hand, but they also create a space of interaction where successful social interactions take, which as we mentioned before is what agent institutions do by setting and enforcing the rules of the game, creating an institutional reality where these rules apply and are enforced. Not surprisingly, agent institutions do have mechanisms that are similar to the ones listed in the description above because institutions (by definition) create the space of opportunity and constrain interactions to better articulate towards the common endeavour. The class of normative MAS and agent institutions are not the same because the mapping between the ideal mechanisms and the way an agent institution framework captures the mechanism is not obvious and is seldom fully established. The following sections will give substance to this last claim but some three prior qualifications are due.

- It is usually assumed that norms ought to be expressed as deontic formulas with a standard proof-theoretic notion of consequence associated to them. This is useful for a declarative description of conventions that is easy to communicate, promulgate and perhaps reason about (at design time as well as at run time). However it is not absolutely necessary, this because there may be other convenient ways of expressing different types of norms. For example, an artificial institution may express conventions that constrain agent actions in procedural (non-declarative) form, for instance using commitment-based protocols and dialogical games, and still use, say, model-checking devices to prove normative properties of the protocol. Likewise, an electronic institution describes permissions, obligations and prohibitions through finite state machines whose transitions are in fact conditional statements in a first order language and paths and propagation take the function of the modal operator; and in these networks, colored Petri nets may provide appropriate semantics for on-line and off-line normative conflict detection, for example.

- It is usually understood that such deontic formulas are enough to fully specify and govern a multi-agent system. Not really. In addition to a collection of norms, a normative MAS requires several institutional constructs in order to legislate, apply, enforce and modify norms. Constitutive conventions for example may need extra-normative devices like bonds and identity certificates to provide entitlements to participating agents. Govenance mechanisms may require the existence of institutional agents that perform norm-enforcement functions, etc.
- Normative notions are pertinent only if norms may be violated. The actual situation is richer. There are application contexts where governance may need to be fully regimented (in electronic markets, for instance) and others that may not (conflict resolution, for example). Hence, enforcement mechanisms in an agent institution may involve a variety of components dealing with observability of actions, institutional power, law enforcement roles, reparatory actions, etc.

Institutions vs organisations The notions of *institution* and *organisation* are closely related. The essential distinction, bluntly speaking, is that the institution is focused on what can be done, while organisations on who does it. Institutions, thus deal mainly with norms and governance, while organisations involve individuals, resources, goals. An institution creates a virtual environment, an organisation is an entity in the world (a crude physical reality). An organisation has boundaries that establish a clear differentiation: some rules apply inside, others apply outside; there is organisational staff, and there are customers and suppliers; there is a macroeconomic environment and there are objectives of the firm. On the other hand the organisation also has several institutional components: best practices, social structure and roles, decomposable activities, internal governance. Although the distinction exists and may be formally stated in a crisp way, when we treat agent institutions, we tend to bundle together the specification of the institution with the implementation of that specification and what really blurs the distinction, we tend to identify the electronic institution (the virtual environment) with the running system that deals with actual transactions: that is, with the computational system *and* the firm that runs it.

Institutional Frameworks In this chapter from Section 2.2 to Section 2.5 we will present three frameworks for agent-based institutions that illustrate how the previously mentioned ideas about institutions are made precise enough to model actual institutions and implement them as multi-agent systems. Those frameworks are: (i) ANTE, a model that considers electronic institutions as computational realizations of adaptive artificial environments for governing multi-agent interactions; (ii) OCeAN extended in MANET, a model for specifying Artificial Institutions (AIs), situated in agent environments, which can be used in the design and implementation of different open interaction systems; and (iii) a framework for Electronic Institutions (EIs), extended with the EIDE development environment, based on open, social, decomposable and dialogical interactions. In Section 2.6 we discuss and compare those three frameworks for agent-based institutions. Finally in Section 2.7 some open challenges in the field of specifications and use of institutions for the realization of real open multi-agent systems are discussed.

We should mention that in addition to these three frameworks, there are at least three other proposals that share the above principles. The first is the *OMNI* model [18], which derives from the *OperA* and *HARMONIA* frameworks introduced in the dissertations of Virginia Dignum [17] and of Javier Vázquez-Salceda [56] respectively. The *OMNI* model allows the description of MAS-based organisations where agent activities are organized as agent scripts (scenes) that are built around a collective goal. The admissible actions of each scene are regulated by a set of norms. The *OMNI* model contains three types of institutional component: normative, contextual and organisational; whose contents are specifiable in three levels of abstraction: descriptive, operational, implementation. Lately, they have developed the *OperettA* framework [1], to support the implementation of real MAS. The second one is the *instAL* framework that puts together the research developed over may years in the University of Bath [15, 13]. *InstAL* is a normative framework architecture and a formal mathematical model to specify, verify and reason about norms that are used to regulate an open MAS. Finally, the third one is the recent proposal by J. Pitt et al. [48] that stems from [5] and draws on institutional notions proposed by E. Olstrom [47].

2.2 The ANTE framework: Electronic Institutions as Dynamic Normative Environments

In this section we will consider electronic institutions as computational realizations of adaptive artificial environments for governing multi-agent interactions.

The use of an *Electronic Institution* as an infrastructure that enables regulation in multi-agent systems presupposes the existence of a common environment where norms (see Part **??**) guide the way agents should behave. The role of an *institutional normative environment* [37], besides providing a set of regulations under which agents' collective work is made possible, is twofold: to check whether agents are willing to follow the norms they commit to (through monitoring), and further to employ correction measures as a means of coercing agents to comply (through enforcement) (see also Chap. **??** on this).

Furthermore, when addressing open systems, the normative environment should enable the run-time establishment of new normative relationships, which are to be appropriately monitored and enforced. Hence, instead of having a predefined normative structure, the shape of the environment will evolve and adapt to the actual normative relationships that are established.

In order to make this feasible, we believe it is important to provide some infrastructure that facilitates the establishment of norm governed relationships. For that, we propose the provision, in an electronic institution platform, of a supportive and extensible *normative framework* [38]. Its main aim is to assist software agents in the task of negotiating and establishing electronic contracts.

Having in mind real-world domains such as agreements guided by electronic contracting, the normative environment will, while monitoring the compliance to norms that apply to specific contracts, record a mapping from the relevant interactions that take place (which concern electronic contracting exchanges). The connection between real-world interactions and the institutional environment is made through illocutions (speech acts) that empowered agents [34] perform with the intent of informing the institution that certain contract-related events have occurred. With an appropriate interface between the normative environment and the statements that agents make, we incrementally build a state of *institutional reality* [51], which is an image of relevant real-world transactions that are, through this means, institutionally recognized (i.e., transactions are turned into *institutional facts* inside the normative environment).

Hierarchical normative framework. In order to facilitate the establishment of electronic contracts, the normative environment should provide a supportive and extensible normative framework. This framework may be inspired by notions coming from contract law theory, namely the use of "default rules" [16] – background norms to be applied in the absence of any explicit agreement to the contrary. We therefore propose that this normative structure is composed of a hierarchy of *contexts* [39], within which norms are created that may apply to sub-contexts. The context hierarchy tries to mimic the fact that in business it is often the case that a B2B contractual agreement forms the business context for more specific contracts that may be created. Each contract establishes a new context for norm applicability.

A *norm defeasibility* approach [38] is also proposed in order to determine whether a norm should be inherited, for a specific situation, from an upper context. This feature allows the normative framework to be adapted (to better fit a particular contract case) and extended (allowing new contract types to be defined). Furthermore, the rationale behind the possibility of overriding any norm is based on the assumption that "default rules" should be seen as facilitating rather than constraining contractual activity [35] (see also Chap. **??** on defeasibility of rules in law).

Adaptive norm enforcement. Adaptive enforcement mechanisms are important in open environments, where the behaviour of an agent population cannot be directly controlled. When the normative specification of contracts includes flaws, namely by omitting normative consequences for some contract enactment outcomes, self-interested agents may try to exploit their potential advantage and intentionally violate contract clauses.

In general, an institution may employ two basic kinds of sanctions in order to incentive norm compliance. Direct *material sanctions* inflict immediate penalties, whereas indirect *social sanctions* have a more lasting effect, e.g. by affecting an agent's reputation. The effectiveness of these alternatives may differ according to the agents that interact within the institutional environment. If agents are not able to take advantage of reputation information, the use of material sanctions is probably a better alternative. Having in mind the deterrence effect of sanctions (i.e., their role in discouraging violations), an institution may use an adaptive sanction model to maintain order (by motivating agents to comply) and consequently trust in the system.

Economic approaches to law enforcement suggest analyzing sanctions by taking into account their effects on parties' activities. Based on this understanding, we have designed and experimentally evaluated a model for *adaptive deterrence sanctions* [40] that tries to enforce norm compliance without excessively compromising agents' willingness to establish contracts. Raising deterrence sanctions has a side effect of increasing the risk associated with contracting activities.

We believe that our approach, which has been implemented as part of the ANTE framework [41], has the distinctive features of being both an open and a computationally feasible approach to the notion of artificial institution. In fact, an *institution* is grounded on some notion of regulation, which is materialized through rules and norms. While some researchers, mostly from fields other than computer science, take an abstract and immaterial perspective to institutions, we find it natural, when addressing electronic institutions, to follow a more proactive stance and ascribe to an electronic institution the role of putting its regulations into practice. These regulations are seen as evolving according to the commitments that agents, when interacting in an open environment, are willing to establish amongst themselves, relying on the institutional environment for monitoring and enforcement purposes. The guiding line for our approach has been the field of electronic contracting.

2.3 The OCeAN metamodel for the specification of Artificial Institution

OCeAN (Ontology CommitmEnts Authorizations Norms)[30, 27] is a metamodel that can be used for specification of Artificial Institutions (AIs). Those institutions thanks to a process of contextualization in a specific application domain can be used and re-used in the design of different *open systems* thought for enabling the interaction of autonomous agents. The fundamental concepts that need to be specified in the design of artificial institutions are:

- an *ontology* for the definition of the concepts used in the communication and in the regulation of the interaction. With an application independent component with concepts and properties that are general enough (like the notion of time, action, event, obligation, and so on) and an application dependent part;
- the possible *events, actions, institutional actions and events* that may happen or can be used in the interaction among agents, this mainly in terms of preconditions that need to be satisfied for their successful performance and effects of their performance;
- the *roles* that the agents may play during an interaction and the rules for playing such roles;
- an *agent communication language (ACL)* for enabling a communication among agents, for example for promising, informing, requesting, agreeing and so on;
- the set of *institutional powers* for the actual performance of institutional actions;
- the set of *norms* for the definition of *obligations*, *prohibitions*, and *permissions*.

In our past works we have proposed a commitment-based semantics of an agent communication language [26] that is regulated by the basic institution of language [30]. We have formalized the concepts for the specification of AIs using different formalisms, and we have used them for specifying the institutions necessary for the design of different types of electronic auctions. In particular initially we specified our metamodel with a notation inspired by the UML metamodel and we used the Object Constraint Language [45] as notation for expressing constraints [31]. Subsequently, due to difficulties of efficiently matching the norms that regulate agents interaction with the actions performed by the agents and the need to perform automatic reasoning on the content of messages and norms, we decided to formally specify the basic concepts of our metamodel by using the Discrete Event Calculus (DEC), which is a version of the Event Calculus. The Event Calculus is a formalism that fits well for the purpose of reasoning about action and change in time, it has has been introduced by Kowalski and Sergot in 1986 [36]. DEC has been introduced by Mueller [42] to improve the efficiency of automated reasoning by limiting time to the integers. This formalism has the advantage of making easier the simulation of the dynamic evolution of the state of the interaction and making possible to perform automated reasoning on the knowledge about the state of the interaction. The main limits of this approach are that the DEC formalism is not widely known among software engineers and the performances of the prototype that we implemented for simulating a run of the English Auction did not scale well with the size of the concepts represented and the number of participating agents.

Consequently in 2009 we started to investigate the possibility to specify our model using Semantic Web Technologies [28, 25] (see also Part **??**). We proposed to specify the concepts (classes, properties, and axioms) of the OCeAN metamodel using OWL 2 DL: the Web Ontology Language recommended by W3C, which is a practical realization of a Description Logic system known as *SROIQ(D)*. We proposed an *upper level ontology* for the definition of the abstract concepts used in the specification of every type of artificial institution, like the concept of *event*, *action*, *time event, change event*, *temporal entity, instant of time* and so on. In particular for modelling time we used the standard OWL Time Ontology[1] enriched with some axioms useful for deducing information about instant of time and intervals. We specified the *OWL Obligation Ontology* [25] that can be used for the specification of the obligations that one agent has with respect to another agent to perform one action that belongs to a class of possible actions, within a given deadline, if certain activation conditions hold, and certain terminating conditions do not hold. Those obligations can be used to specify constrains on the behaviour of the interacting agents and to express the semantics of conditional promises communicative acts [29]. The *OWL Obligation Ontology* together with some functionalities realized for performing closed world reasoning a certain classes can be used for *monitoring* the evolution in time of the state of the obligations on the basis of the events and actions that happens during the interaction. In fact reasoning in OWL is based on an *open world assumption* but in our model, in order to be able to deduce that an obligation to perform an action, when the deadline is elapsed, is violated, we need to implement closed-world reasoning and assuming that in the interaction contexts where this model will be used, not being able to infer that action has been performed in the past is sufficient evidence that the action has not been performed. Regarding monitoring it is also important to solve the problem of finding an efficient and effective mechanism for mapping real agents' actions in element of the OWL ontology for being able to perform automated reasoning on them and deducing that an obligation to perform a given action is fulfilled or violates. Currently the OCeAN meta-model has not been completely specified using Semantic Web Technologies, we plan to do it in our future works.

The main advantage of the choice of using Semantic Web technologies is that they are increasingly becoming a standard for Internet applications, and given that the OWL logic language is decidable, it is supported by many reasoners (like Pellet and HermiT), tools for ontology editing (like Protégé) and library for automatic ontology management (like OWL-API and JENA). Moreover the specification of artificial institutions in OWL makes them easily reusable as data construct in many different applications in different domains.

2.4 Artificial Institutions Situated in Environment: the MANET model

Thanks to the Agreement Technology COST Action in 2009 we started to investigate how to integrate the studies on the model of agent environments [57], in particular the model presented in the GOLEM framework [10], with the OCeAN meta-model of AI. As first result of this work we proposed the MANET (Multi-Agent Normative EnvironmenTs) model where AI are situated in agent environments [54].

One of the most important tasks of an *environment* is to mediate the actions and events that happen, where *mediate* means that an environment is in charge of regis-

[1] http://www.w3.org/TR/owl-time/

tering that an event has happened and of notifying this event to all agents registered to the template of this event (the agents that have a sensor for this type of events) [10]. An environment is composed of *objects* and *physical spaces*, and is the place where *agents* interact. A physical space describes the infrastructure of the system and its infrastructural limitations to the agents behaviour in terms of physical rules.

Given that AIs are abstract description specified at design time, it is crucial to specify how certain AI can be concretely used at run-time for the definition and realization of open systems. Therefore we proposed to introduce in the model of environments the notion of *institutional space* that is used for having a first-class representation of AIs. In particular institutional spaces represent the boundaries of the effects of institutional events and actions performed by the agents, they may contain sub-spaces, and they enforce the norms of the system in response to the produced events.

Given that institutional spaces may contain sub-spaces, it is possible that the different AIs, used for the specification of different institutional spaces, may present some interdependencies. For example in a marketplace we can have many different auctions represented with sub-spaces created using different AIs. Given that agents may contemporarily participate in more than one space, it may happen that the norms of one space, for example the marketplace, regulate also some events of its sub-spaces, for example by prohibiting to an agent to do bid in an auction represented in a sub-space if it has a specific role in the market-place. For soling this problem it is necessary to give to the designer of the system the possibility to define events that may be *observed* outside the boundaries of the space. Another problem may arise when the rules a space (for example an auction) regulate for instance the participation of an agent to another space (another auction or a contract). In this case we need to introduce in the model the possibility for one space to *notify* another space about the fact that a specific event is happened.

The MANET model of artificial institutions situated in environment has been implemented in Prolog on top of GOLEM platform [10] and it was used for formalizing and running an e-energy marketplace [54] where agents representing different types of energy producers try to sell energy to potential consumers.

2.5 Electronic Institutions

The work we have been doing in the IIIA on electronic institutions (EIs, for short) may be observed from four complementary perspectives:

1.*The mimetic perspective:* EIs can be seen as computational environments that mimic the coordination support that conventional human institutions provide.

2. *The regulated MAS perspective* understands EIs as open multi-agent systems, that organise collective activities by establishing a restricted virtual environment where all interactions take place according to some established conventions.

3. *EIs as "artifacts" perspective* takes EIs to be the operational interface between the subjective decision-making processes of participants and the social task that is achieved through their interactions.

4. *The coordination support perspective:* EIs are a way of providing structure and governance to open multi-agent systems.

These four characterizations are supported by one single abstract model whose assumptions and core components we briefly discuss below. In turn, as we'll also see below, this abstract model is made operational through a set of software components that follow one particular computational architecture.

Over the past few years we have had the chance to build numerous examples of electronic institutions in a rather large variety of applications with those tools [19][2].

A conceptual core model for Electronic Institutions. Electronic institutions are grounded on the following basic assumptions about interactions:

- *Open.* Agents are black-boxes, heterogenous, self-motivated and may enter and leave the institutional space on their own will.
- *Social.* Agents come together in pursuit of an endeavour that requires a collective participation; thus agents need to be aware of other agents and their roles and of the capabilities needed to achieve a particular goal in a collective activity.
- *Decomposable.* To contend with the possibility (due to openness) of large number of agents being involved in the social interaction we allow the collective endeavour to be decomposed into atomic activities (*scenes*) that achieve particular goals with the participation of fewer individuals. The decomposition requires that scenes be connected in a network in which the achievement of individual and collective goals correspond to paths in that network.
- *Replicable.* Simple activities may be either re-enacted by different groups of agents or enacted concurrently with different groups.
- *Co-incident.* An agent may be active, simultaneously, in more than a single activity[3].
- *Contextual.* Openness and decomposability limit the knowledge agents have of each other, thus interactions are naturally *local* within subgroups of agents that share a common "scene context", while as a dynamic virtual entity, the collectivity of agents is itself immersed in a larger "institutional context".
- *Dialogical.* Activities are achieved through interactions among agents composed of non-divisible units that happen at discrete points in time. Thus construable as point-to-point messages in a communication language, so that even physical actions may be thus wrapped[4].

---

[2] The IIIA model of Electronic Institutions is the result, mainly, of three dissertations [43, 50, 20]

[3] We will deal with to this ubiquity of a given agent as *agent processes* that stem from it, so that we have an objective ground for concurrency and control issues when implementing the institutional infrastructure.

[4] Messages make reference to an application domain and should be properly "anchored" (their meaning and pragmatics should be established and shared by participants), e.g. the term "pay" entails the real action of transferring funds in some agreed upon way; in a trial, the constant "exhibit A" corresponds to some object that is so labeled and available at the trial.

These assumptions allow us to represent the conventions that will regulate agent interactions with the few constructs depicted in Fig. 2.1. The full detail of these constructs is presented in [3] but, broadly speaking, to specify an EI we need:

1. *A dialogical framework* that consists essentially of (i) a social model of roles and their relationships; (ii) a domain and a communication languages that will be used to express the institutional messages, plus a few other languages for expressing institutional constraints, and (iii) an information model to keep the *institutional state*, that is, the updated values of institutional variables.
2. *A performative structure* that captures the high level structure of the institutional interactions as a network of scenes connected by transitions.
3. *Procedural and behavioural constraints* that affect the contents of the performative structure; namely, (i) preconditions and postconditions of messages within scenes, (ii) constraints on the movement of roles between scenes and (iii) propagation of the effects of actions among scenes; for expressing all these constraints we make use of the tower of languages of the dialogical framework.

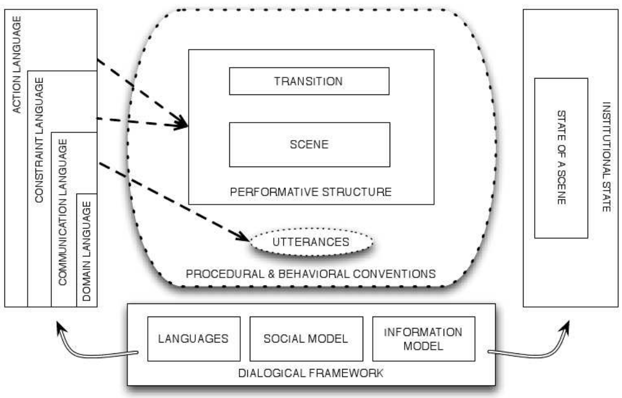


Fig. 2.1 Sketch of the Electronic Institutions Conceptual Model.

Our model has a straightforward operational semantics: institutional reality is changed through agent actions, but only those agent actions that comply with the institutional constraints have any institutional effect. More precisely, the institutional state is only altered through actions that comply with the procedural and behavioural constraints and in our model the only possible actions an agent can take are: to utter a message, to enter and leave the institution, and to move between scenes. Figure 2.1, hides the fact that an *electronic* institution also constitutes the infrastructure that *enables* actual interactions. Thus, we need that our conceptual model includes all those operations that need to be supported by the infrastructure; namely, those operations triggered by the actions of an agent that we just mentioned, plus those operations that the infrastructure itself needs to accomplish so that the first ones are feasible. Table 2.1 summarizes all those operations, the last column indicates the constructs that the operation updates.

Operation	Called by	Effect on
Speak	Agent	scene
RequestAccess	Agent	electronic institution
JoinInstitution	Agent	electronic institution, scene
LeaveInstitution	Agent	electronic institution, scene
SelectNewTargets	Agent	transition
RemoveOldTargets	Agent	transition
StartElectronicInstitution	Infrastructure	electronic institution
CreateSceneInstance	Infrastructure	scene institution
CloseSceneInstance	Infrastructure	scene
EnableAgentsToLeaveOrTransition	Infrastructure	transition
EnableAgentsToLeaveAndTransition	Infrastructure	transition
MovingFromSceneInstanceToTransitionInstance	Infrastructure	scene, transition
MoveAgentFromTransitionToScene	Infrastructure	scene, transition
RemoveClosedInstances	Infrastructure	electronic institution
Timeout	Infrastructure	scene

Table 2.1 Electronic institution operations

One computational architecture for Electronic Institutions. The model just presented may be implemented in different ways. We have chosen one particular architecture (see [24]) where we build a centralized institutional infrastructure that is implemented as a separate "social milieu" that mediates all the agent interactions, as Fig. 2.2 shows.

- *Governor* All communications between a given agent and the institution are mediated by a corresponding infrastructure agent that is part of the institutional infrastructure called the *governor* (indicated as G in Fig. 2.2).[5] The governor keeps a specification of the institution plus an updated copy of the institutional state, thus when its agent produces an utterance, that utterance is admitted by the governor if and only if it complies with the institutional conventions as they are instantiated at that particular state; only then, the utterance becomes an institutional action that changes the state. Likewise, the governor communicates to the agent those institutional facts that the agent is entitled to know, the moment they happen. Additionally, the governor controls navigation of its agent between scenes, and the production of new instances of the agent itself (*agent processes*). It also keeps track of time for synchronization (time-outs) purposes. Note that in order to provide these services, a governor must coordinate with scene managers, transition managers, and the institution manager. In this realisation of the

[5] Agents cannot interact directly with one another, they use an agent communication language (like JADE) to interact with their governors who mediate their interactions *inside* the electronic institution.

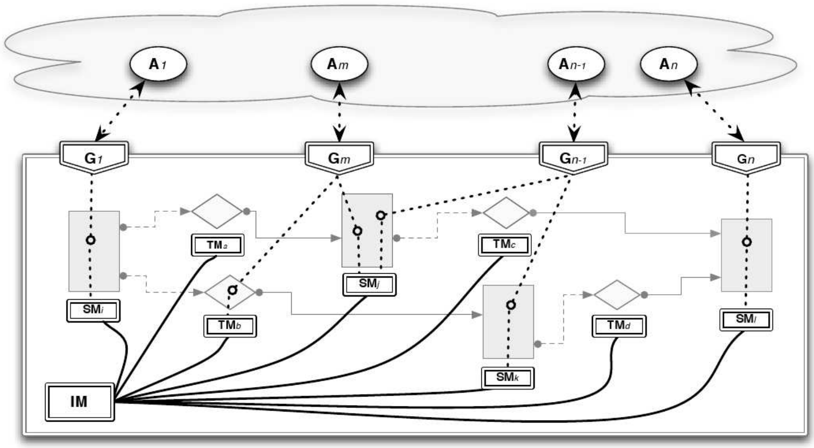


Fig. 2.2 An architecture for electronic institutions. Participating agents (A), communicate with (infrastructure) governor agents (G), which in turn coordinate with other infrastructure manager agents for each scene (SM) and each transition (TM) and with the institution manager agent (IM).

EI framework, therefore, governors are involved in the implementation of most of the operations in Table2.1.

- *Institution Management* Each institution has one *institution manager* agent (IM), which activates (*StartElectronicInstituion* operation) and terminates the institution. It also controls the entry (*RequestAccess*, *JoinInstitution*) and exit (*LeaveInstitution*) of agents, together with the creation the closing of scenes (*CloseSceneInstance*, *RemoveClosedInstances*). Finally, it keeps track of the electronic institution state.
- *Transition management* Each transition has a *transition manager* (TM) that controls the transit of agents between scenes by checking that requested moves are allowed (*EnableAgentsToLeaveOrTransition*, *EnableAgentsToLeaveAndTransition*) and, if so, allowing agents to move (*MovingFromSceneInstanceToTransitionInstance*, *MoveAgentFromTransitionToScene*).
- *Scene management* Each scene has an associated infrastructure agent, the *scene manager* (SM), who is in charge of: starting and closing the scene (in coordination with the institution manager); keeping track of agents that enter and leave the scene; updating the state of the scene by processing utterances (*Speak*) and time-outs (*Timeout*); and coordinating with transition managers to let agents in or out of a scene (*MovingFromSceneInstanceToTransitionInstance*, *MoveAgentFromTransitionToScene*).

Other architectures are feasible and we have, for instance, suggested a peer-to-peer variant of these ideas in [23].

A development environment based on that architecture. The computational model we just described, does not commit to any specific convention about the languages used in the specification of transitions and scenes, nor on the syntax and pragmatics of illocutions, nor on specific governance mechanisms. Those commitments come later when software tools to build actual electronic institutions become implemented. One way of implementing the computational model is the Electronic Institutions Development Environment (EIDE) [22] which includes the following tools:

ISLANDER: a graphical specification language, with a graphic interface [21]. It allows the specification of any EI that complies with the conceptual model and produces an XML file that the AMELI middleware runs[6].

AMELI: a software middleware that implements the functions of the social layer at run-time [24]. It runs an enactment, with actual agents, of any ISLANDER-specified institution. Thus it activates infrastructure agents as needed; controls activation of scenes and transitions, access of agents, messages between agents and institution, and in general guarantees —in coordination with infrastructure agents— the correct evolution of scenes and the correct transitions of agents between scenes. AMELI may be understood as a two-layered middleware. One public layer formed by governors, the other private layer —not accessible to external agents— formed by the rest of the infrastructure agents. External agents are only required to establish communication channels with their governors[7]. Infrastructure agents use the institutional state and the conventions encoded in the specification to validate agent actions and evaluate their consequences.

SIMDEI: is a simple simulation tool used for debugging and dynamic verification. It is coupled with a *monitoring tool* that may be used to display the progress of the enactment of an institution. It monitors every event that takes place and may display these events dynamically with views that correspond to events in scenes and transitions or events involving particular agents. Both tools may be used for dynamic verification.

aBUILDER: an agent development tool which, given an ISLANDER-specified institution, supports the generation of "dummy agents" that conform to the role

[6] ISLANDER allows static verification of a specification. It checks for *language integrity* (all roles and all terms used in illocutions, constraints and norms are properly specified in the dialogical framework), *liveness* (roles that participate in a given scene have entry and exit nodes that are connected and may be traversed), *protocol accessibility* (every state in the graph of a scene is accessible from the initial state and arcs are properly labeled), *norm compliance* (agents who establish "normative commitments" may reach the scenes where the commitments are due). ISLANDER may be extended to have a strictly declarative expression of scene conventions [33].

[7] The current implementation of the infrastructure can either use JADE or a publish-subscribe event model as communication layer. When employing JADE, the execution of AMELI can be readily distributed among different machines, permitting the *scalability* of the infrastructure. Notice that the model is communication-neutral since agents are not affected by changes in the communication layer.

specification and are able to navigate the performative structure, provided agent designers fill up their decision-making procedures[8].

2.6 Conclusions: A comparison of the described institutional models

In this section we compare the three proposed models of institutions, ANTE, OCeAN/MANET, and EI, discussing their crucial differences and analogies on a set of relevant aspects.

- *Institutional reality*.
  All three models adhere to the representation of institutional reality proposed by John Searle in [51], in particular on the existence of an institutional reality that has a correspondence with the real or physical world, and on distinguishing between "institutional" facts and actions, on one side, and their possibly corresponding "brute" facts and actions, on the other.

- *Social model: roles and hierarchy of roles*
  - ANTE accommodates two types of roles within the institution. Agents providing core institutional services are seen as performing *institutional roles* that are under the control of the institution. Agents acting as delegates of external entities enact different roles that are normatively regulated by the institution, in the sense that they may be subject to norms and may further establish new normative relationships. Furthermore, some of these roles are empowered, through appropriate *constitutive rules*, by the institution to ascertain institutional reality (i.e. they act as trusted third parties from the institution's point of view).
  - OCeAN/MANET allows the definition of roles as labels defined by a given Artificial Institution (AI) and used in the AI to assign norms and institutional powers at design time to roles. This is necessary because at design time the name of the actual agents that will take part to the interaction is unknown. At run time AIs are realized in dynamically created institutional spaces, the agents in a space can start to play the roles defined in the space and coming from the AI. An agent can play more than one role contemporarily. During an interaction an agent can start to play a role and subsequently stop to play it.
  - EI allows for specification of role subsumption and the specification of two forms of compatibility among roles: "dynamic" (each agent may perform different roles in different activities) and "static" no agent may perform both roles in an enactment of the institution. It also distinguishes between *internal*

[8] Based on the same ideas, there is an extension of aBUILDER [9] that instead of code skeletons produces a simple human interface that complies with the ISLANDER specification and is displayed dynamically via a web browser at run-time.

roles (played by agents whose behaviour is controlled by the institution), and *external* roles (the institution has no access to their decision-making capabilities) and this separation is static.

- *Atomic interactions*
  - ANTE, concerning its institutional component, assumes an open setting in which there are two kinds of interactions going on in the system. On one hand, agents are free to interact with any other agents, without the institution even noticing that such interactions have taken place. On the other hand, illocutionary actions performed by agents towards the normative environment are seen as attempts to obtain *institutional facts* that are used by the latter to maintain the normative state of the system.
  - OCeAN/MANET defines *institutional actions* that in order to be successfully performed needs to satisfy certain conditions. One of these conditions is that the actor of the action needs to have the power to perform the institutional action, otherwise the action is void. The model defines also *instrumental actions*, for example the exchange of messages that should be used to perform institutional actions. Finally in the model it is possible to represent actions performed in the real world and that are relevant for the artificial interaction, for example the payment of an amount of money or the delivery of a product.
  - EI: There are essentially only two types of institutional actions: *speech* acts (represented as illocutions) and the *movement* actions which are accomplished in two steps exiting from a scene to a transition and entering from a transition to a scene (in some contexts an agent may *stay-and-go*, i.e remain active in the scene while at the same time becoming active in one or more different scenes)[9]. Consequently, on one side, an agent can act only by uttering an illocution or notifying the institutional environment its intention to move in or out of a transition (possibly changing role); on the other side, the perception of any given agent is restricted to those illocutions that are uttered by another agent and have the given agent as part of the intended listeners of that illocution, and the indication of the institutional infrastructure that a movement has been achieved

- *Institutional state*
  - ANTE: The institutional normative state is composed of two sorts of so-called *institutional reality elements*. *Agent-originated events* are obtained as a consequence of agent actions, comprising essentially *institutional facts* that are obtained from the illocutions agents produce. These institutional facts map relevant real-world transactions that are through this means institutionally recognized. *Environment events*, on the other hand, occur as an outcome of the process of norm triggering and monitoring. Norms prescribe directed obligations with time windows, which when monitored may trigger different enact-

[9] In fact, as indicated in Table 2.1 these movements are implemented with five operations, which include the two key actions of entering and leaving the electronic institution.

ment states, namely temporal or actual violations, and fulfilments. All these elements are contextualized to the normative relationships that are established within the environment.
- OCeAN/MANET: In the last version of the model the state of the interaction is represented using OWL 2 DL ontologies, one of the international standard language of the Semantic Web. Therefore the state of the interaction is represented using classes of concepts, individuals that belong to classes, object and data properties that connect two individuals or an individual to a literal (scalar values) respectively. The terminological box of the ontologies is also enriched with axioms, used to describe the knowledge on a given domain of application, and with SWRL rules, both are used by software reasoners to deduce new knowledge on the state of the interaction. Taking inspiration from the environment literature the state of objects, agents, events, and actions in a space are perceivable by the agents in that space.
- EI: Only atomic interactions that comply with the institutional regimented conventions may be institutional actions and therefor change institutional facts. There is a data structure called the *institutional state* that contains all the institutional facts; that is, all the constants in the domain language plus the updated values of all those variables whose values may change through institutional actions. For each scene there exists a projection of that structure called the *state of the scene*. Additionally, there are some parameters whose default values are set by the institution and may be updated during an enactment. These are *institutional variables* (like the number of active scenes, the labels of active scenes and transitions), *scene variables* (like the number of participants, the list of items that remain to be auctioned, performance indicators such as the number of collisions or the rate of successful agreements) and *agent variables* (the list of external agents that have violated any discretional convention, the credit account of a trader). These parameters are not accessible to external agents although by design they may be accessible to some internal agents who may use the values of these variables in their individual decision-making.

- *Structure of the activities or compound interactions (contexts)*

  - ANTE: Interactions that need to be observed are executed through empowered agents, which will then inform the institutional environment of the actual real-world activities that are taking place. Such activities are segmented into different normative contexts, that is, they pertain to specific normative relationships that are established at run-time. Within each such context different empowered agents may need to act as intermediaries, since different kinds of actions may need to be accomplished in order to successfully enact the contract subsumed in the context.
  - OCeAN/MANET: The activities are realized into *institutional spaces* or *physical spaces* of interaction. Institutional spaces are used to realize AI at runtime, they may be entered and left by the agents starting from the root space. Physical spaces contains physical entities external with respect to the system,

such as external resources, databases, external files, or web services, offering an abstraction that hide the low level details from the agents. Institutional spaces are in charge of representing and managing the social interaction of agents by realizing the concepts described in AIs and the services for norms monitoring and enforcement. Spaces are in charge of registering that an event has happened and represents the boundaries for the perception and of the effects of the events and actions.

- EI: Activities are decomposable into *scenes* that are connected by *transitions* into a network of scenes called a *performative structure*.
  - Scenes are state transition graphs where edges are labeled by *illocutionary formulas* and nodes correspond to a scene-state. A new scene-state may only be attained with the utterance of an admissible illocution. An utterance is valid if and only if it complies with the regimented conventions that apply under the current state of the scene. At some scene-states agents may enter or leave or *stay-and-go* the scene. Every performative structure contains one “start” and one “finish” scene that have the merely instrumental purpose to delimit the structure for syntactic (in specification) and implementation purposes (for enactment of the electronic institution).
  - A transition is a device that is used for two main purposes, to control role flow and to control causal and temporal interdependence among scenes. In particular, (a) when an agent exits a scene, it exits with the role it was playing in that scene but inside the transition the agent may change that role to enter a new scene (provided some institutional conventions are satisfied) (b) Moreover, when an agent enters a transition and depending on the type of transition it enters, that agent may join one, several or all the scenes that are connected to that transition. (c) Several agents, possibly performing different roles and coming possibly from different scenes, may enter the same transition and each has to decide on its own where to go from there and wether it changes role or not. (d) The transition coordinates flow by determining whether agents may proceed to their intended goal scene as soon as each agent arrives or wait until some condition holds in the state of the scene.

- *Hierarchical organisation of the structure of activities*

  - ANTE: Normative relationships established at run-time are organized as a hierarchy of contexts. Each context encompasses a group of agents in a specific regulated organisation, within which further sub-contexts may be created, allowing for norm inheritance to take place. An overall institutional normative layer is assumed to exist, of which every subsequently created context is a sub-context. Furthermore, each context may add its own norms, which may be used to inhibit norm inheritance or to enlarge the normative framework that will govern the context.
  - OCeAN/MANET: Spaces may contain other spaces generated dynamically at run-time, which become sub-spaces of the space where they are created. This hierarchy of spaces and the fact that one agent may be simultaneously in two

spaces create interesting problems due to the interdependencies of spaces, this because the events of a space may be of interest to the father-space where this is contained or for a sibling space.
- EI: All agent interactions within an electronic institution are organized, as we mentioned above, by what we call a performative structure which is a network of scenes and transitions between those scenes. Two aspects are worth stating: First, a performative structure may be be embedded into another as if it were a scene, thus forming nested performative structures of arbitrary depth. Second, a performative structure becomes instantiated at run-time, thus although it is defined *a priori*, so to speak, the actual scenes do not come into existence until appropriate conditions take place (if ever) and they disappear likewise. In particular, it is possible to specify conditions that empower an internal agent to spawn a particular scene or performative (sub)structure.

• *Procedural and functional conventions*

- ANTE: The effects of institutional facts are expressed through norms and rules. When triggered, norms prescribe directed obligations that are due to specific agents within a normative context. Such obligations have attached time-windows that are conventionally understood as ideal time periods for obtaining the obliged state of affairs. Outside this window temporal violations are monitored which may lead to different outcomes depending on the will of the obligation's counterpart. This semantics is captured by a set of monitoring rules that maintain the normative state of the system. The normative consequences of each obligation state is determined by the set of norms that shape the obligation's normative context, which may be established at run-time.
- OCeAN/MANET: Both are expressed through pre and post-conditions of the actions defined by the institution. An important pre-condition for the performance of institutional actions (actions whose effects change institutional attributes that exist only thanks to the collective acceptance of the interacting agents) is the fact that the actor of the action should have the institutional power to perform the specific action.
- EI: Both are expressed as pre and post-conditions of the illocutionary formulas of the scene transition graphs and through the labeling of transitions between scenes (this labeling expresses conditions for accessing a scene or a group of scenes or a nested performative structure, synchronization, the change of roles, the creation of new scenes or activation of an existing scene). In the current EIDE implementation, there is also the possibility of explicitly expressing norms as production rules that are triggered whenever an illocution is uttered, thus allowing the specification and use of regimented and not-regimented conventions. Notice that although EI use illocutionary formulas to label actions, there are no social semantics of illocutionary particles involved. Thus scene protocols are not commitment-based protocols as is the case with [26] or more generally, [14, 12].

- *Constitutive conventions*
  - ANTE: Obtaining institutional facts from brute facts (which are basically agent illocutions) is achieved through appropriate *constitutive rules*, which mainly describe empowerments of different trusted third parties. These constitutive rules, which can be easily extended and/or adapted, determine the ontology for brute and institutional facts that can be used in the institution. Furthermore, it is possible to define further constitutive rules within each context, in this case enriching the domain ontology by obtaining more refined institutional facts. As a basic implementation, three types of transactions are reportable to the normative environment, related with the flow of products, money and information.
  - OCeAN/MANET: In this model the content language used for communicative acts and norms is defined using domain ontologies written in OWL 2 DL or in RDF+RDF Schema. Those ontologies may be defined by the designer of the interaction system or may already exist as proposed standards on the Web, like the well known ontology FOAF[10] that may be used for describing agents. In many cases the link between the name of a resource (its URI) and the corresponding resource in the real world can be done using existing knowledge repositories[11].
  - EI: The EI framework does not include axioms or definition statements that establish basic institutional facts. Nevertheless, there is a *domain language* that is used for expressing illocutionary formulas and whose terms correspond with physical facts and actions (e.g a sculpture to be auctioned, pay 32 euros for the item that has just been adjudicated). The correspondence between language and real entities is established *ad-hoc* for the domain language. In practice, however, an electronic institution needs to have true constitutive conventions in order to establish the legal (actual) entitlements of intervening parties and the correspondence between institutional and brute facts and actions. Examples of constitutive conventions are the contracts that allow an old books dealer to offer a used book through Amazon.com and follow the process through from offer to book delivery.

- *Social Commitments*
  - ANTE: Social commitments, in a broad sense, are established as an outcome of a previous negotiation phase, the success of which obtains a new normative context within the institutional environment. Once a normative context is obtained, applicable norms dictate when (according to the normative state) and which commitment instantiations (directed obligations) are entailed.
  - OCeAN/MANET: A commitment-based Agent Communication Language (ACL) is used [26, 29]. In particular communicative acts exchanged among agents have a meaning that is a combination of the meaning of the content of

[10] `http://www.foaf-project.org/`
[11] `http://linkeddata.org/`

the messages and a meaning of the illocutionary force of the communicative acts (for example promise, query, assert).

– EI: although, in EI, illocutionary formulas label actions, there is no social semantics of the illocutionary particles involved. Thus scene protocols are not commitment-based protocols properly speaking. However, commitments are hard-wired in scene specifications , and their evolution is captured in the evolving state of the institution. It should be noted, though, that in EI some commitments are expressed crudely but explicitly when a given admissible action (say winning a bidding round) has a postcondition that entails preconditions for future actions in other scenes.

- *Governance*
  – ANTE: The approach adopted in ANTE is to bear with the autonomy of agents, by allowing them to behave as they wish. From the institution's perspective, we assume it is in the best interest of agents to publicize their abidance with any standing obligations, by using the necessary means to obtain the corresponding institutional facts. Normative consequences of (non)fulfilment are assured by triggering applicable norms. Permissions and prohibitions are not handled explicitly in the system, i.e., not permitted actions simply have no effect within the normative environment. Entitlements are handled by defining norms triggered upon the occurrence of specific institutional facts. Any obligation outcomes – (temporal) violations and fulfilments – may also have further effects within the ANTE framework by reporting such events to a computational trust engine, which provides a mechanism of indirect social sanctioning.
  – OCeAN/MANET: The openness of the interaction systems realized using this model requires a governance in order to create an expectations on the actions of the participants agents. Contemporarily the model has to take into account the autonomy and heterogeneity of the interacting agents and avoid to constrain their behaviour in rigid protocols. The main concepts introduced in the model related to governance are: *institutional power* (if an agent has not the power to perform an action its effects are void), *permission* (if an agent has not the permission to perform an action its effect take place but the agent incur in a violation), *obligations* (the agent has to perform an action with-in a given deadline) and *prohibitions* (the agent cannot perform an action, if it does it will incur in a violation).
  – EI: There are three different approaches for the implemention of governance in the EI model.
    1. In the standard model, all regimented conventions may be encoded in the performative structure as part of the specification of scenes and transitions and are therefore enforced in a strict and automatic fashion by the runtime implementation. Non-regimented conventions are encodable in the decision-making capabilities of internal agents and it is a matter of design whether some regimented ones may also be embedded in internal agents code. One may thus establish different types of (internal) norm-

enforcement agents. Notice that although an internal agent may fail or decide not to enforce a violation, every violation is observed (registered) by the institution nonetheless.
2. In the current implementation of EIDE one may choose to specify a collection of normative statements that are not part of the performative structure. This collection is coupled with an inference engine that takes hold of every utterance before it may be validated by the performative structure (see [33]). The process is as follows (i) An illocution is first tested against the normative statements and if it is consistent, it is labeled as "admissible" or rejected otherwise. (ii) The admissible illocution is then added to the current collection and the engine is activated; (iii) If the illocution triggers a violation, the concomitant corrective actions are taken, otherwise control is given to the performative structure that deals with the illocution as in approach 1. This approach allows to deal with discretional enforcement with more flexibility than approach (1) because in addition to all the mechanisms available in that approach, this one allows for a declarative specification of norms, an explicit distinction between regimented and non-regimented norms, and a variety of contrary-to-duty devices encodable as corrective actions.
3. There is a proposed extension of the EI model that deals explicitly with norms and normative conflicts through the use a a "normative structure" that deals exclusively with norms and propagation of normative consequences between scenes [32, 55].

- *Ubiquity and concurrent activities*
  - ANTE: Agents may freely establish new normative relationships, and hence many of them may be active at the same time. The institutional environment pro-actively monitors every active context. There is a strong distinction between the agent identity and the normative relationships in which it is engaged. There is no notion of "physical" displacement of the agents within the institution. Within the ANTE framework, several other activities may take place at the same time, such as negotiations and computational trust building, which is achieved by gathering relevant enactment data from the normative environment monitoring process.
  - OCeAN/MANET: An interaction system realized using one or more AIs consists of a root space that contains physical and institutional spaces. An agent situated in a given space can enter all its sub-spaces, therefore an agent can be in more than one space and it has a persistent identity.
  - EI: An electronic institution usually consists of multiple scenes that are active simultaneously. In many cases the number of active scenes changes during execution since new scenes are created, activated or closed as the enactment proceeds. A given agent may be simultaneously active in more than one scene but it has a persistent identity in the sense that the effects of its institutional actions are coherent (for example, in an electronic market where an agent may be closing deals in different negotiations, this agent has *one* variable that captures

its credit so the value of that variable changes every time it commits to pay, in whatever scene they commit). The current EI framework does not include a "meta-environment" where multiple institutions co-exist, however the peer-to-peer architecture proposed in [23] would be suitable for the implementation of lightly-coupled (and uncoupled) institutions in a shared environment.

- *Performance Assessment*
  - ANTE: Agent performance is assessed and exploited from two different perspectives. The first one is based on computational trust: the enactment of contracts produces evidences that are fed into a computational trust engine, which then produces trustworthiness assessments of agents that can be used when entering into further negotiations. In the current prototype implementation, trust information may be used for pre-selection of negotiation peers or for proposal evaluation. Another assessment of performance is measured by the normative environment, which for the whole agent population is able to determine the average enactment outcome for instances of stereotyped normative relationships (types of contracts).
  - OCeAN/MANET: There are not yet available services for assessing system's or agent performance.
  - EI: This model does not capture system goals explicitly, however scene and institutional variables may be used to specify some assessment of the performance of the institution with respect to whatever goals are defined. Internal agents may be designed to use such information in order to improve performance.
- *Formal properties*
  - ANTE: No formal methods for analyzing normative relationships are employed – it is up to the system designer to ensure correctness. The normative environment does record on-line every possible event that is captured while monitoring norms, allowing for an off-line verification of correctness.
  - OCeAN/MANET: For the moment there is not the possibility to check formal properties of AI at design-time. At run-time one crucial service is the monitoring of the state of the interaction, the detection of violations, and the enforcement of norms. Moreover in every instant of time it is possible to deduce the list of the actions that an agent is obliged, prohibited, permitted and empowered to perform, from this list and from an ontological definition of the terminology used to describe the actions it is possible to single out possible contradiction in the prescribed behaviour. At design time this check is harder because in this model all normative constrains are related to time.
  - EI: There is off-line automatic syntactic checking of scene and transition behaviour. For example, in every scene: all roles have entry and exit states and these are reachable; every role has at least one path that takes it from start to finish; every term used in an illocution needs to be part of the domain ontology. On-line monitoring of all the activities: every utterance and attempted move produce a trace that may be displayed and captured for further use. The

extensions mentioned in [55] allow for some off-line and on-line formal and automated reasoning about an institution.

- *Institutional Dynamics*
  - ANTE: The normative environment is assumed to be open and dynamic, in the sense that it encompasses an evolving normative space whose norms apply if and when agents commit to a norm-governed relationship. While providing an institutional normative framework, this infrastructure enjoys the properties of adaptability and extensibility, by providing support for norm inheritance and defeasibility. Normative contexts can therefore be created that adapt or extend a predefined normative scenario according to agents' needs.
  - OCeAN/MANET: This model is based on the idea that a human designer specifies an AI and this AI may be used at run-time to dynamically create spaces of interaction. Similarly norms at design time are specified in terms of roles and have certain unspecified parameters, at run-time those norms will be instantiated more than one time having as debtor different agents and different values for their parameters. In general this model does not include meta-operations for changing the model of AIs.
  - EI: With the current model internal agents may be given the capability to create new scenes from repositories of available scenes and even graft nested performative structures into a running institution. In a similar fashion internal agents may create new internal agents when needed (say for a newly grafted performative structure) by invoking a service that spawns new agents that is outside of the electronic institution proper but is available to the internal agent. This mechanism is also used to embed the EI environment into a simulation environment [4]. The current model includes no primitive meta-operations that would allow agents to change the specification of an institution beyond what was just said, however here have been proposals for other forms of autonomic adaptation [8, 11].
- *Implementation architecture*
  - ANTE: The ANTE framework is realized as a Jade FIPA-compliant platform, where agents can make use of the available services (e.g. negotiation, contract monitoring, computational trust) through appropriate interaction protocols, such as FIPA-request and FIPA-subscribe. Using subscription mechanisms agents are notified of the normative state of the system in which their normative relationships are concerned. The normative environment has been implemented using the Jess rule-based inference engine.
  - OCeAN/MANET: The model of AI has been fully formalized in Event Calculus and we are currently formalizing it using Semantic Web Technologies. An AI for realizing a Dutch Auction has been also specified in PROLOG and tested in a prototype realized above the GOLEM environment framework [54]. An implementation of a complete energy market-place based on Semantic Web Technologies and the GOLEM framework is under development.

  - EI: The model has been fully detailed [19] in the Z specification language [53] and deployed in the architecture sketched in Fig.2.2. This architecture creates a sort of "social layer" that is independent of the communication layer used to exchange messages between an agent and the electronic institution. The normative engine extension is also implemented in the same architecture. A peer-to-peer architecture has been proposed [23] and a prototype is now under construction.

- *Tools*

  - ANTE: The ANTE framework includes graphical user interfaces (GUI) that allow the user to inspect the outcomes of each provided service, including the evolution and outcome of a specific negotiation, the inspection of trustworthiness scores of the agents in the system, as well as the overall behaviour of the agent population in terms of norm fulfillment. The framework includes also a complex API allowing for the specification of user agents, for which a set of predefined GUI are also available that enable the user to inspect the agent activity, namely its participation in negotiations and contracts. The API allows a programmer to easily encode agent behaviour models in response to several framework activities, such as negotiation and contract enactment, which makes it straightforward to run different kinds of experiments (although Jade has not been designed for simulation purposes).
  - OCeAN/MANET: Thanks to the fact that we base our model of current standard semantic web technologies, it is possible to use the ontology editor Protègè for editing the ontologies used in the specification of the model of AI and spaces and to use one of the available reasoners (Pellet, HermiT, and so on) for checking their consistency. Our future goal is that once the model of a set of AIs is defined and a set of agents able to interact with a system getting its formal specification are developed, the interaction system can start to run and enable agents to interact using the available actions and constrained by the specified norms.
  - EI: As mentioned in the previous section, EIDE includes a graphical specification language (ISLANDER), an agent middleware for electronic institutions (AMELI) that generates a runtime version of any ISLANDER compatible specification. EIDE also includes an automated syntactic checker, a simple simulator for on-line testing and debugging, a monitoring tool, and a software that generates agent skeletons that encode the navigational behaviour that is compatible with an ISLANDER specification.

- *Agents*

  - ANTE: The framework is neutral in which user agents' internal architectures and implementation languages are concerned. It is assumed, however, that agents are able to communicate using FIPA ACL and the FIPA-based interaction protocols and ontologies interfacing each of the framework's services. It is also straightforward to admit human agents to participate, provided that appropriate user interfaces are developed.

- OCeAN/MANET: The model of the interaction system realized using the AI is independent on the agents' internal structure. Nevertheless it is assumed that the participating agents are able to interact using the available communicative acts whose content should be expressed using shared ontologies.
- EI: The model is agent-architecture independent. Agents are required only to comply with interface conventions that support institutional communication. Hence human agents may participate in an electronic institution enactment provided they have the appropriate interfaces. The tool HIHEREI [9] automatically generates such a human interface for any ISLANDER compatible specification of an electronic institution. In the current implementation, AMELI is communication-layer independent.

2.7 Challenges

There are many open challenges in the field of specification and use of institutions for the efficient realization of real open interaction systems in different fields of applications, going from e-commerce, e-government, supply-chain, management of virtual enterprise, and collaborative/socila resource sharing systems.

One interesting challenge goes into the direction of using those formal and declarative models of *hybrid* open interaction systems involving both software and human agents. In this perspective one possibly important use of these technologies is for designing flexible open collaborative/social systems able to exploit the flexibility, the intelligence, and the autonomy of the interacting parties. This in order to improve existing business process automation systems where the flow of execution is completely fixed at design time or groupware where the work of defining the context and the rules of the interaction is left to the human interacting parties and no automatic monitoring of the completion of tasks is provided.

When considering the automation of e-contracting systems through autonomous agents, another important challenge is to endow agents with reasoning abilities that enable them to establish more adequate normative relationships. Infrastructural components need to be developed that ease this task, e.g. through normative frameworks that agents can exploit by relying on default norms that may nevertheless need to be overridden. A complementary challenge is how to ensure reliable behaviours when agents act as human or enterprise delegates, that is, how to simultaneously cope with expressivity and configurability through human interfaces and agents' autonomy in institutional normative environments. Another interesting challenge is to look at the Environment as a structured medium not only to facilitate agents' interaction but also as an active representative of the "society" in which agent relationships take place.

References

1. Aldewereld, H. and Dignum, V. Operetta: Organization-oriented development environment. In M. Dastani, A. E. Fallah-Seghrouchni, J. Hübner, and J. Leite, editors, *LADS*, volume 6822 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2010.
2. Aoki, M. *Toward a Comparative Institutional Analysis*. MIT Press, 2001.
3. Arcos, J.L, Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., and Sierra, C. Engineering open environments with electronic institutions. *Eng. Appl. of AI*, 18(2):191–204, 2005.
4. Arcos, J.L., Noriega, P., Rodriguez-Aguilar, J.A., and Sierra, C. E4mas through electronic institutions. In D. Weyns, H. Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems III.*, number 4389 in Lecture Notes in Computer Science, pages 184–202. Springer, Berlin / Heidelberg, 08/05/2006 2007.
5. Artikis, A., Sergot, M., and Pitt, J. Specifying norm-governed computational societies. *ACM Trans. Comput. Logic*, 10(1):1:1–1:42, Jan. 2009.
6. Boella, G., Pigozzi, G., and van der Torre, L. Five guidelines for normative multiagent systems. In G. Governatore, editor, *Legal Knowledge and Information Systems. JURIX 2009*, pages 21–30, Amsterdam, October 22-24 2009. IOS Press.
7. Boella, G., van der Torre, L., and Verhagen, H. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.
8. Bou, E., Lopez-Sanchez, M., and Rodriguez-Aguilar, J.A. Adaptation of autonomic electronic institutions through norms and institutional agents. In G. O'Hare, A. Ricci, M. O'Grady, and O. Dikenelli, editors, *Engineering Societies in the Agents World VII*, volume 4457 of *Lecture Notes in Computer Science*, pages 300–319. Springer Berlin / Heidelberg, 2007.
9. Brito, I., Pinyol, I., Villatoro, D., and Sabater-Mir, J. HIHEREI: Human Interaction within Hybrid Environments. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)*, pages 1417–1418, Budapest, Hungary, 2009.
10. Bromuri, S., and Stathis, K. Distributed agent environments in the Ambient Event Calculus. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 12:1–12:12, New York, NY, USA, 2009. ACM.
11. Campos, J., Lopez-Sanchez, M., and Esteva, M. A case-based reasoning approach for norm adaptation. In A. S. E. Corchado, M. Graña-Romay, editor, *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS'10)*, volume 6077, pages 168–176, San Sebastian, Spain, 23/6/2010 2010. Springer, Springer.
12. Chopra, A., and Singh, M. Nonmonotonic commitment machines. In F. Dignum, editor, *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Computer Science*, pages 1959–1959. Springer Berlin / Heidelberg, 2004.
13. Cliffe, O., De Vos, M., and Padget, J. Specifying and reasoning about multiple institutions. In P. Noriega, J. Vzquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organization, Institutions and Norms in Agent Systems II - AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006 Hakodate, Japan, May 9, 2006 Riva del Garda, Italy, August 28, 2006*, volume 4386 of *Lecture Notes in Computer Science*, pages 67–85. Springer Berlin / Heidelberg, 2007.
14. Colombetti, M. A commitment-based approach to agent speech acts and conversations. In *Proceedings of the Fourth International Conference on Autonomous Agents, Workshop on Agent Languages and Conversation Policies.*, pages 21–29, 2000.
15. Corapi, D., De Vos, M., Padget, J., Russo, A., and Satoh, K. Normative design using inductive learning. *Theory and Practice of Logic Programming*, 11:783–799, 2011.
16. Craswell, R. Contract Law: General Theories. In B. Bouckaert and G. De Geest, editors, *Encyclopedia of Law and Economics*, volume Volume III: The Regulation of Contracts, pages 1–24. Edward Elgar, Cheltenham, UK, 2000.
17. Dignum, V. *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, University Utrecht, 2004.
18. Dignum, V., Vázquez-Salceda, J., and Dignum, F. OMNI: Introducing social structure, norms and ontologies into agent organizations. In *PROMAS*, volume 3346 of *Lecture Notes in Computer Science*, pages 181–198. Springer, 2004.

19. d'Inverno, M., Luck, M., Noriega, P., Rodriguez-Aguilar, J.A., and Sierra, C. Communicating open systems. *Artificial Intelligence*, 186(0):38 – 94, 2012.
20. Esteva, M. *Electronic Institutions: from specification to development. PhD thesis Universitat Politècnica de Catalunya (UPC), 2003.* Number 19 in IIIA Monograph Series. IIIA, 2003.
21. Esteva, M., de la Cruz, D., and Sierra, C. ISLANDER: an electronic institutions editor. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent systems (AAMAS '02)*, pages 1045–1052. ACM Press, July 2002.
22. Esteva, M., Rodriguez-Aguilar, J.A., Arcos, J.L., Sierra, C., Noriega, P. and Rosell, B. Electronic Institutions Development Environment. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '08)*, pages 1657–1658, Estoril, Portugal, 12/05/2008 2008. International Foundation for Autonomous Agents and Multiagent Systems, ACM Press.
23. Esteva, M., Rodriguez-Aguilar, J.A., Arcos, J. L., and Sierra, C. Socially-aware lightweight coordination infrastructures. In *AAMAS'11 12th International Workshop on Agent-Oriented Software Engineering*, pages 117–128, 2011.
24. Esteva, M., Rodriguez-Aguilar, J.A., Rosell, B., and Arcos, J. L. AMELI: An agent-based middleware for electronic institutions. In C. Sierra and L. Sonenberg, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS '04)*, volume 1, pages 236–246, New York, USA, July 19-23 2004. IFAAMAS, ACM Press.
25. Fornara, N. Specifying and monitoring obligations in open multiagent systems using semantic web technology. In A. Elçi, M. T. Kone, and M. A. Orgun, editors, *Semantic Agent Systems: Foundations and Applications*, volume 344 of *Studies in Computational Intelligence*, chapter 2, pages 25–46. Springer-Verlag, 2011.
26. Fornara, N. , and Colombetti, M. Operational specification of a commitment-based agent communication language. In *Proceedings of The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy*, pages 536–542. ACM, 2002.
27. Fornara, N. , and Colombetti, M. Specifying Artificial Institutions in the Event Calculus. In V. Dignum, editor, *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, Information Science Reference, chapter XIV, pages 335–366. IGI Global, 2009.
28. Fornara, N. , and Colombetti, M. Representation and monitoring of commitments and norms using OWL. *AI Communications - European Workshop on Multi-Agent Systems (EUMAS) 2009*, 23(4):341–356, 2010.
29. Fornara, N. , Okouya, D. and Colombetti, M. Using OWL 2 DL for expressing ACL Content and Semantics. In M. Cossentino, M. Kaisers, K. Tuyls, and G. Weiss, editors, *EUMAS 2011 Selected and Revised papers*, LNCS, page to appear, Berlin, Heidelberg, 2012. Springer-Verlag.
30. Fornara, N. , Viganò, F. , and Colombetti, M. Agent communication and artificial institutions. *Autonomous Agents and Multi-Agent Systems*, 14(2):121–142, April 2007.
31. Fornara, N. , Viganò, F. , Verdicchio, M. , and Colombetti, M. Artificial institutions: A model of institutional reality for open multiagent systems. *Artificial Intelligence and Law*, 16(1):89–105, March 2008.
32. Gaertner, D. , Garcia-Camino, A. , Noriega, P. , Rodriguez-Aguilar, J. A., and Vasconcelos, W. W. Distributed norm management in regulated multi-agent systems. In *Proceedings of the 6th international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07)*, pages 624–631, Honolulu, Hawaii, 14/05/07 2007. ACM Press.
33. Garcia-Camino, A. , Noriega, P. and Rodriguez-Aguilar, J. A. Implementing norms in electronic institutions. In S. T. Michal Pechoucek, Donald Steiner, editor, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*, pages 667–673, Utrecht, NL, 2005. ACM Press.
34. Jones, A. J. I., and Sergot, M. J. A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, 4(3):427–443, 1996.

35. Kaplow, L. General Characteristics of Rules. In B. Bouckaert and G. De Geest, editors, *Encyclopedia of Law and Economics*, volume Volume V: The Economics of Crime and Litigation, pages 502–528. Edward Elgar, Cheltenham, UK, 2000.
36. Kowalski, R. A. and Sergot, M. J. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.
37. Lopes Cardoso, H. *Electronic Institutions with Normative Environments for Agent-based E-contracting*. PhD thesis, Universidade do Porto, 2010.
38. Lopes Cardoso, H., and Oliveira, E. Norm defeasibility in an institutional normative framework. In M. Ghallab, C. Spyropoulos, N. Fakotakis, and N. Avouris, editors, *Proceedings of The 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 468–472, Patras, Greece, 2008. IOS Press.
39. Lopes Cardoso, H. and Oliveira, E. A Context-based Institutional Normative Environment. In J. Hubner, E. Matson, O. Boissier, and V. Dignum, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, LNAI 5428, pages 140–155. Springer, 2009.
40. Lopes Cardoso, H., and Oliveira, E. Social control in a normative framework: An adaptive deterrence approach. *Web Intelli. and Agent Sys.*, 9:363–375, December 2011.
41. Lopes Cardoso, H., Urbano, J., Rocha, A., Castro, A., and Oliveira, E. ANTE: Agreement Negotiation in Normative and Trust-enabled Environments. In *Handbook of Agreement Technologies (Chapter on Applications), in this volume*. Springer, 2012.
42. Mueller, E. T. *Commonsense Reasoning*. Morgan Kaufmann, San Francisco, 2006.
43. Noriega, P. *Agent-Mediated Auctions: The Fishmarket Metaphor. PhD thesis Universitat Autònoma de Barcelona, 1997*. Number 8 in IIIA Monograph Series. IIIA, 1999.
44. North, D. C. *Institutions, institutional change, and economic performance*. Cambridge University Press, Cambridge, 1990.
45. Object Management Group. UML 2.0 OCL Specification. http://www.omg.org/, 2005.
46. Ostrom, E. An agenda for the study of institutions. *Public Choice*, 48(1):3–25, 1986.
47. Ostrom, E. Institutional analysis and development: Elements of the framework in historical perspective. In C. Crothers, editor, *Historical Developments and Theoretical Approaches in Sociology in Encyclopedia of Life Support Systems(EOLSS), Developed under the Auspices of the UNESCO*. Eolss Publishers, Oxford, UK, 2010.
48. Pitt, J., Schaumeier, J., and Artikis, A. Coordination, conventions and the self-organisation of sustainable institutions. In *Proceedings of the 14th international conference on Agents in Principle, Agents in Practice*, PRIMA'11, pages 202–217, Berlin, Heidelberg, 2011. Springer-Verlag.
49. Powell, W. W. and Dimaggio, P. J. *The New Intitutionalism in Organizational Analyisis*. University of Chicago Press, 1991.
50. Rodriguez-Aguilar, J. A. *On the Design and Construction of Agent-mediated Electronic Institutions, PhD thesis, Universitat Autònoma de Barcelona, 2001*. Number 14 in IIIA Monograph Series. IIIA, 2003.
51. Searle, J. R. *The construction of social reality*. Free Press, New York, 1995.
52. Simon, H. A. *The Sciences of the Artificial*. MIT Press, third edition, 1996.
53. Spivey, J. M. *Understanding Z: A Specification Language and its Formal Semantics*. Cambridge University Press, 1988.
54. Tampitsikas, C., Bromuri, S., Fornara, N., and Schumacher, M.I. Interdependent Artificial Institutions In Agent Environments. *Applied Artificial Intelligence*, 26(4):398–427, 2012.
55. Vasconcelos, W., García-Camino, A., Gaertner, J. A. Rodríguez-Aguilar, and P. Noriega. Distributed norm management for multi-agent systems. *Expert Systems with Applications*, 39:5990–5999, 04/2012 2012.
56. Vázquez-Salceda, J. *The role of Norms and Electronic Institutions in Multi-Agent Systems applied to complex domains. The HARMONIA framework*. PhD thesis, Universidad Politecnica de Catalunya, 2003.
57. Weyns, D., Omicini, A., and Odell, J. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007.

Chapter 3
Organisational Reasoning Agents

Olivier Boissier and M. Birna van Riemsdijk

Abstract Regarding the agent's architecture perspective and analysing the reasoning capabilities of agents with respect to organisations, different cases may be considered: agents may have or not have an explicit representation of the organisation, and they may be able to reason on it or not. Organisational reasoning agents have the capability to represent the organisation and are able to reason on it. In this chapter, we will discuss the main features of this kind of agents and which are the fundamental mechanisms for reasoning on organisations. We will also describe some approaches proposed in the literature related to how agents can take decisions on their participation in an organisation.

3.1 Introduction

In a MAS, agents are situated in a common environment, and are capable of flexible and autonomous behaviour. They make use of different cognitive elements and processes in order to control their behaviour (e.g. beliefs, desires, goals, capacities of situation assessment, of planning). Their autonomy is among the most important characteristics of the concept of agency. However, this autonomy can lead the overall system to exhibit undesired behaviour, since each agent may do what it wants. This problem may be solved by assigning an organisation to the system, as it is done in human societies. Roles as they are defined in organisational models, are generally used to flag the participation of an agent to the organisation and to express what the expected behaviour is of that agent in the organisation. In the literature, more or less formal specifications of the requirements of a role exist (see for instance [1]

Olivier Boissier
FAYOL-EMSE, LSTI, Saint-Etienne, France e-mail: Olivier.Boissier@emse.fr

M. Birna van Riemsdijk
Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft, The Netherlands e-mail: m.b.vanriemsdijk@tudelft.nl

on the different notions of roles and [15]). Combined with the different dimensions that are expressed in the organisational models supporting the organisation specification, this leads to different sets of constraints that can be imposed to the agent's behaviour while participating to an organisation (constraints on beliefs, on goals, on the interaction protocols that it can use while cooperating with other agents, on the agents to communicate with, etc).

From this global picture at the macro level (i.e. organisation perspective), let's have a look at the micro level, i.e. agent perspective. Taking an agent's architecture perspective and analysing the reasoning capabilities with respect to organisation, different cases may be considered [22, 2]: first, agents may have or not have an explicit representation of the organisation, and second, they may be able to reason on it or not. In this section, we mainly consider agents that, internally, have the capability to represent the organisation and that are able to reason on it. They could consider the organisation as a help to decide what to do (e.g., coalition formations [33]), and/or as a set of constraints that aim at reducing their autonomy or, on the contrary may help them to gain certain powers.

From what precedes, one could ask the reason why it would be worth having such kind of agents in a multi-agent organisation. From the analysis drawn in [3], mainly from human societies, it clearly appears that when an agent plays a role, its behaviour and its cognitive elements and processes change. Correspondingly, one may want to *recreate* these kinds of processes also when artificial agents play roles in artificial organisations.

Moreover, agents that are able to reason about organisations are needed in order to realize *open systems* [4, 20]. Increasingly, it is recognized that the internet (including latest developments into sensor networks and 'internet of things') can form an open interaction space where many heterogeneous software agents co-exist and act on behalf of their users. Such open systems need to be regulated. However, such regulation is only effective if agents can understand the imposed regulations and adapt their behaviour accordingly, i.e., if agents are capable of organisational reasoning.

Finally, organisational reasoning agents facilitate engineering multi-agent systems adhering to the principle of *separation of concerns*. That is, when agents can reason about an organisation, the agents and the organisation can be developed separately. When the system designer changes parts of the organisation, e.g., norms that agent playing a certain role should adhere to, one does not need to change the agents as they will be able to adapt (within reasonable limits) to the changed organisation.

There are different ways in which an agent's cognitive elements or behaviour can change because of the role it plays. It may adopt role's goals, desires or beliefs, it may acquire knowledge or new powers. It may also acquire or lose some powers and finally it may decide to do what's best for the organisation, putting aside (for the moment) its own goals. Any agent playing a role is faced with the problem of integrating the cognitive elements of the role with its own. Moreover, when the internal motors of the agent change, its behaviour is likely to change too. An agent should also change its way of reasoning, to cope with the new dimensions of its behaviour, i.e., its mental processes are different when it plays a role. Besides the changes on the individual dimension of an agent, playing a role affects also the agent's relationships with other agents: change of the agent's status by interpreting all of the agent's physical actions, communications, beliefs, etc. as being the ones of its role, acquisition/loss of powers, dependence relationships with respect to other agents, trust relationship by being more (or less) trusted by others, etc.

After this brief introduction sketching the motivations for having organisation aware agents, we will first present in section 3.2, some fundamental mechanisms for reasoning on organisations, identifying how and what kind of organisation primitives agents may have. We will then present some approaches proposed into the literature that illustrate the use of reasoning on organisation. The adaptation of organisations being addressed in the following chapter (cf. Chap. 4), we focus here on the kind of reasoning that an agent should develop for the entry/exit in/of an organisation (cf. In section 3.3) considering both the ability and desirability points of view.

3.2 Mechanisms for Reasoning on Organisations

In order to be able to develop reasoning behaviours on the organisation, an agent must be equiped with fundamental mechanisms as described in a very abstract way in Fig. 3.1 [31]. The agent must be equipped with a basic set of primitives to act on the organisation and, the dual aspect, the capabilities to acquire the organisation description and represent it internally. Then it should be able to reason with this representation, affecting the agent's cognitive reasoning (reasoning about how to achieve goals and react to events).

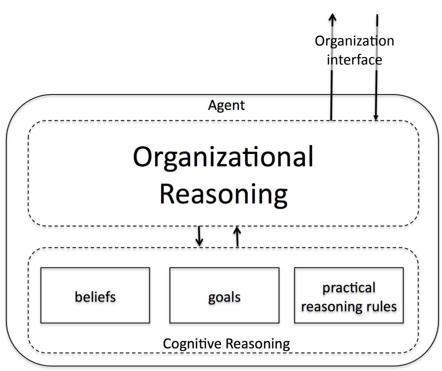


Fig. 3.1 Abstract description of Organisational Reasoning Agent Architecture [31]

These capabilities must be included in an agent architecture for reasoning about the different constructs induced by the participation of the agent to an organisation.

Different concrete architectures have been proposed (e.g. [11, 7, 26, 25]). Each of these allows agents to represent and reason about various treatments of norms and organisations.

3.2.1 Mechanisms for making agents aware of the organisation

Several proposals have been made in the literature, dealing with the way agents are connected to the organisation, i.e. how agents acquire the description of the organisation (either abstract specification of it or concrete one in terms of which agent plays what, etc). To illustrate this more clearly, let's consider the $\mathscr{M}$OISE organisational model (explained in Chap. 1 Sect.1.2 of this book) for which there is available an extension of the Jason language [5] to develop reasoning plans and strategies on the organisation. This extension allows developers to use this high-level BDI language to program agents able to reason on the organisation, by making them able to acquire organisational descriptions, especially its changes (e.g., a new group is created, an agent has adopted a role), and to act upon it (e.g., create a group, adopt a role). In this model, the way it is done is strongly connected to the set of organisational artifacts [24] that instruments the MAS environment to support the management of the organisations expressed with the $\mathscr{M}$OISE organisation model.

These different concrete computational entities aimed at managing, outside the agents, the current state of the organisation in terms of groups, social schemes, and normative state encapsulate and enact the organisation behaviour as described by the organisation specifications.
From an agent point of view, such organisational artifacts provide the actions that can be used to proactively take part in an organisation (for example, to adopt and leave particular roles, to commit to missions, to signal to the organisation that some social goal has been achieved, etc.). They dynamically also provide specific observable properties to make the state of an organisation perceivable to the agents along with its evolution, directly mapped into agents' percepts (leading to beliefs and triggering events). So as soon as the observable properties values change, new percepts are generated for the agent that are then automatically processed (within the agent reasoning cycle) and the belief base updated. Besides, they provide actions that can be used by agents to manage the organisation itself (sanctioning, giving incentives, reorganising). They provide the operations and the observable properties for agents so that they can interact with the organisation. This means that - at runtime - an agent can do an action α if there is (at least) one artifact providing α as operation - if more than one such artifact exist, the agent may contextualise the action explicitly specifying the target artifact. We refer the interested reader to [24, 25] to have a look at the available repertoire of actions and observable properties.

So in programming an agent it is possible to write down plans that directly react to changes in the observable state of an artifact or that are selected based on contextual conditions that include the observable state of possibly multiple artifacts.

3.2.2 Mechanisms for organisational reasoning

Development of mechanisms for full-fledged organisational reasoning is still in its early stages. Nevertheless, several approaches have been proposed, some of which we briefly describe below.

The following papers address role enactment. In [17] an approach is proposed in the context of agent programming that defines when an agent and a role match or are conflicting. An agent can enact a role if they are not conflicting. Enactment is then, broadly speaking, specified as taking up the goals of the role, and defining a preference relation over the agent's own goals and the role's goals. In [18] the authors propose programming constructs that allow an agent to enact and deact a role. The semantics of the constructs is defined by specifying how the agent's mental attitudes change when a role is enacted/deacted. In [30] it is investigated how agents can reason about their capabilities in order to determine whether they can play a role (see also Section 3.3.1). It is shown how reasoning about capabilities can be integrated in an agent programming language.

Once an agent enacts a role, it should take into account the norms and regulations that come with the role in its reasoning. In [27], an approach is proposed on how AgentSpeak(L) agents can adapt their behaviour to comply with norms. Algorithms are provided that allow an AgentSpeak(L) agent to adopt goals upon activation of obligations, or remove plans upon activation of prohibitions. Even if an agent participates in an organisation, it may still decide to violate some of the corresponding norms. In [28] it is investigated how to extend plans with normative constraints that are used to customize plans in order to comply with norms. In [6] an approach based on prioritized default logic is proposed, that allows to express whether an agent prioritizes obligations, desires or intentions. Based on this prioritization, the agent generates the goals that it will pursue. In [11] an architecture is proposed by means of which norms can be communicated, adopted and used as meta-goals on the agent's own processes. As such they have impact on deliberation about goal generation, goal selection, plan generation and plan selection. The architecture allows agents to deliberatively follow or violate a norm, e.g., because it has a more important personal goal. Another proposal for deliberation about norms is put forward in [16]. It investigates the usage of coherence theory in order to determine what it means to follow or violate a norm according to the agent's mental state and making a decision about norm compliance. Moreover, consistency notions are used for updating agent mental state in response to these normative decisions. In [14], an extended BDI reasoning architecture is proposed for 'organisationally adept agents' that balances organisational, social, and agent-centric interests and that can adjust this balance when appropriate. Agent organisations specify guidelines that should influence individual agents to work together in the expected environment. However, if the environment deviates from expectations, such detailed organisational guidelines can mislead agents into counterproductive or even catastrophic behaviours. The proposed architecture allows agents to reason about organisational expectations, and adjust their behaviours when the nominal guidelines misalign with those expectations. In [29] norms are taken into account during an agent's plan generation phase. Norms can be obligations or prohibitions which can be violated, and are accompanied by repair norms in case they are breached. Norm operational semantics is expressed as an extension/on top of STRIPS semantics, acting as a form of temporal restrictions over the trajectories (plans) computed by the planner.

3.3 Reasoning on the participation to an organisation

In this section we will see different approaches related to entering an organisation, playing a role in the organisation and leaving the organisation. Agents should be able to decide whether to enter an organisation, consider whether they are able to participate and whether they really desire to participate; and we will also analyse how roles affect agents, i.e., how playing a role affects directly an individual and how playing a role affects an individual's relationships with others.

3.3.1 Am I able to participate to an organisation?

An important aspect that organisational reasoning agents should be able to reason about is whether they are able to play a role in an organisation, i.e., about whether it has the required *capabilities* [30].

This is important as it allows an agent to decide, e.g., only to apply for roles for which it has (some of) the capabilities. Also, an agent may have to communicate about the capabilities that it has. For example, consider organisations in which a dedicated agent (a *gatekeeper*) is responsible for admitting agents to the organisation. An example of an organisational modelling language in which such a gatekeeper is present, is OperA [19]. The idea is then that the gatekeeper asks agents who want to join whether they have the necessary capabilities for playing the desired role in the organisation (similar to a job interview), and assigns roles to agents on the basis of this. In order to be able to answer the gatekeeper's questions, the agent needs to know what its capabilities are.

In order to develop general techniques that allow agents to determine what their capabilities are, it is important to make precise what kind of capabilities are considered. One may consider various capability types, like capabilities to execute *actions*, to *perceive* aspects of the environment in which the agents operate, to *communicate* information, questions or requests, and to achieve *goals* [30].

Once it is precisely defined which capability types are considered, the agent should be endowed with mechanisms that allow it to *reflect on its own capabilities*. Reflection can in general be seen as an agent's introspective abilities. Reflection is also a technical term in programming. It allows a program to refer to itself at run-time (see, e.g., Java and Maude [13]), which facilitates a modification of its run-time behaviour based on these reflections. Reflection in the latter sense can be a way to implement an agent's introspective abilities. In [30] it was proposed to allow an agent to derive beliefs about its capabilities, in this way integrating reflection in a natural way in its BDI reasoning mechanisms.

3.3.2 Do I desire to participate to an organisation?

Besides being able to detect if it is able to play a role in an organisation, it is also necessary for an agent to detect if it is worth to be part of an organisation.

For instance, in [8], social commitments and social policies have been used to express what an agent is expected to do when entering an organisation. As in [34] where playing a role is considered as a contract, it is considered that an agent playing a role in an organisation implies a set of commitments towards the organisation in which it plays this role. A role is thus defined by the social commitments it implies, but also by the resources put at the disposal in order to fulfil the social commitments that come with the role. We can classify the constraints imposed to an agent playing a role in an organisation into several categories:

- goals to achieve: when it accepts to play a role, an agent accepts to try to achieve several goals, the role's goals.
- authority relations: a role can have authority over another goal for something.
- context-dependent obligations: when playing a role, an agent might have to fulfil several obligations towards the organisations.
- permissions and prohibitions: when it accepts to play a role, an agent receives permissions to do some tasks and prohibitions to do others.

From that understanding, the agent translated these commitments into power relations on which it was able to install social-power reasoning mechanisms that it used before deciding whether to adopt a role or not in order to assess the implications of this decision, i.e. what it will gain or lose by playing the role, what changes are likely to occur in his reasoning or behaviour.

This analysis and classification on the playing of a role may be conducted along two main directions: how playing a role affects directly an individual, how playing a role affects an individual's relationships with others.

3.3.2.1 How playing a role affects directly an individual

There are different ways in which an agent's cognitive elements or behaviour change because of the role it plays. It may adopt role's goals, desires or beliefs, it may acquire knowledge or new powers. It may also acquire or loose some powers and finally it may decide to do what's best for the organisation.

Adoption of role's goals, desires, beliefs: Most of related work in MAS focuses on the need for an agent to adopt the desires or goals of its role: most formal organisations divide the global goal of the organisation into subgoals delegated to its members, which are identified by the roles they play. Since role's goals can facilitate or hinder the achievement of agents' own set of goals (Dastani et al., 2003), agent adoption of role's goals may depend on:

- degree of autonomy, internal motivations. If there is no conflict between the role's and the agent's goals, then an agent will adopt its role's goals and will try to pursue them. If there is a conflict and the goals cannot be satisfied together, an agent should choose what to do: (i) it could either not adopt the role's goals, (ii) it could adopt them and discard its own contradicting goals, (iii) it could adopt all the goals and make a decision later which of its currently contradicting goals will pursue
- organisational incentives, etc.

Acquisition of knowledge, of new powers: In order to ensure that its members are able to achieve their roles' goals, an organisation usually: gives these members access to sources of information or knowledge, trains them to better perform their tasks, gives them physical resources (money, a house, a car, etc.) or permissions to access and use organisation's resources. Autonomous agents accept to wear a role because of the acquisition of: knowledge, access to information, new powers [9] (using the resources coming with role and associated permissions). However, agents might use knowledge/power for their own interest or they can take advantage from an information source (e.g., a library) or power to satisfy their own personal goals.

Losing powers: When an agent agrees to wear a role in a group, it signs a more or less formal or explicit contract with the group: what powers will be given to the agent (resources, permissions) and lost by the agent (prohibitions, obligations), which of his powers an agent puts at the disposal of the group.

Role's prohibitions are one of the reasons for losing powers: If an agent was able to satisfy a goal, it will not be able anymore if there is a prohibition to pursue that goal or to execute a key action in the plan to achieve that goal. playing a role might imply the agent loses the physical access to a resource.

Role's obligations hinder an agent's powers in a more subtle way: by obliging the agent to consume resources needed for other goals.

Putting powers at the disposal of a group means that the agent's decision process is no longer autonomous: his decision process is influenced (or even controlled) by an external entity. He thus loses other powers because he is no longer free to decide to use them.

Desire the best for the group: Agents, even if self-interested, usually desire the best for the organisation they belong to: often implicit in an agent (especially in the case of MAS), but is behind many decisions made by the agent when playing a role in that group.

Importance in multi-agent organisations to make explicit not only a role's goals and norms, but also this desire.

Agents' behaviour affected in many ways when playing these roles: using their personal powers for the best of the organisation, by enabling a functional violation of norms [10], i.e. to violate norms if it's in the organisation's best interest.

This desire to ensure the best of the group should be present in all roles and agents should adopt it when playing these roles. It might affect agents' behaviour in many ways, like using their personal powers for the best of the organisation, but also by enabling a functional violation of norms [10]. Agents could decide to disobey the norms imposed to their roles if they believe that by doing this they increase the well-being of the organisation. We believe that is important in multi-agent organisations to make explicit not only a role's goals and norms, but only this desire with its high importance, thus enabling agents to violate norms if it's in the organisation's best interest.

3.3.2.2 How playing a role affects an individual's relationships with others

Playing a role may impact the relationships an agent develop with other agents in different ways, in term of status, powers, dependence relationships and/or trust.

Count-as effect: playing a role changes the agent's status: all of its physical actions, communications, beliefs, etc. are interpreted as being the ones of its role, e.g. other agents interpret executed actions/communication as being the role that executed the action/communication, and not the agent (e.g. command has a different meaning coming from a role with authority or from a simple agent). Importance for agents to have a means to express whether their actions, communications, ... count as the actions, communications, ... of their role or not. Agents should be aware of this and act accordingly. This limits the ways they can behave.

Acquisition/losing powers: Roles in an organisation belong to a rich network of relationships that are inherited by the agents playing the roles. e.g. authority relationship : a "superior" role has authority over an "inferior" role for something, meaning that whenever an agent playing the superior role delegates a goal (or an action, etc.) to an agent playing an inferior role, the latter must adopt and achieve it. These relationships modify the powers of an agent playing a role: an agent playing a role with authority over another gains a power over the agent playing the inferior role, i.e. the first agent disposes whenever it wants of one of the powers of the second agent (the power for which it has authority). The first agent thus gains an indirect power, while the second agent loses its power, by losing the possibility to decide about it. The higher the role of an agent in the role hierarchy, the more indirect powers it gains: however, due to the relative nature of authority, an agent could have power over others for something, while the others will have power over it for something else.

Dependence relationships: Even in a non-organisational context, when not playing any role, agents depend on each other for a power and not for another power [33]: lack of power of achieving goals, lack of the needed resources or know-how. Not only agents have dependence networks, but also roles in organisations [21]: agents playing the roles inherit these relationships and usually must use the role's dependence network instead of their own.

An agent should not solve only conflicts between his goals, beliefs, etc., and the ones of his role, but also conflicts between his personal dependences and those of his role. An interesting situation occurs when an agent wears several roles in the same time and must put together and use several dependence networks, situation from which an agent might benefit sometimes.

Being more (or less) trusted by others: Trust relationships [32] between agents change when they wear roles (see Part. **??**). Institutional trust [12] : An agent can be trusted by others simply because it plays a role in an institution. The others' trust in it comes from their beliefs in the characteristics of the role inherited by the agent. Another reason to trust more an agent playing a role in a group is because the group acts as an enforcer: there are incentives for an agent to obey the role's specifications.

3.4 Conclusions

Organisations represent an effective mechanism for activity coordination, not only for humans but also for agents. Nowadays, the organisation concept has become a relevant issue in the multi-agent system area, as it enables analysing and designing coordination and collaboration mechanisms in an easier way, especially for open systems. In this section we have presented some works in the direction aiming at endowing the agents with reasoning capabilities on organisations. We have focused on the kind of reasoning that agents should develop on entering or not in an agent organisation. In the current landscape of agreement technologies this is an important issue in the sense that the systems that are considered are large scale and open systems. We can also add to this kind of reasoning, all the different reasoning methods developed for organisation adaptation (described in the next chapter), for norm compliance, given the fact that norms are often considered in the context of organisations (see Part **??**). Besides these different reasoning, we have also described basic and fundamental mechanisms that make agents able to develop these different kinds of reasoning.

References

1. Boella, G., Odell, J., van der Torre, L., Verhagen, H.: Roles, an interdisciplinary perspective, papers from the 2005 AAAI fall symposium. Technical Report FS-05-08, The AAAI Press, Menlo Park, California (2005)
2. Boissier, O.: Modèles et architectures d'agents. In: Briot, J.P., Demazeau, Y. (eds.) Principes et Architectures des Systèmes Multi-Agents, pp. 71–107. IC2, HERMES (Nov 2001)
3. Boissier, O., Carabelea, C., Castelfranchi, C., Sabater-Mir, J., Tummolini, L.: The dialectics between an individual and his role. In: Boella, G., Odell, J., van der Torre, L., Verhagen, H. (eds.) Roles, An Interdisciplinary Perspective, Papers from the 2005 AAAI Fall Symposium. vol. Technical Report FS-05-08, pp. 13–18. The AAAI Press, Menlo Park, California (2005)

4. Boissier, O., Hübner, J.F., Sichman, J.S.a.: Organization oriented programming: from closed to open organizations. In: Proceedings of the 7th Int. Conference on Engineering societies in the agents world VII. pp. 86–105. ESAW'06, Springer-Verlag, Berlin, Heidelberg (2007)
5. Bordini, R., Hübner, J., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. John Wiley & Sons, Ltd (2007)
6. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly 2(3-4), 428–447 (2002)
7. Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., van der Torre, L.W.N.: The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In: Agents. pp. 9–16 (2001)
8. Carabelea, C., Boissier, O., Castelfranchi, C.: Using social power to enable agents to reason about being part of a group. In: Proceedings of 5th International Workshop on Engineering Societies in the Agents World (ESAW'04). LNCS, vol. 3451, pp. 166–177. Springer (2005)
9. Castelfranchi, C.: The social nature of information and the role of trust. Int. J. Cooperative Inf. Syst. 11(3), 381– (2002)
10. Castelfranchi, C.: Formalising the informal? Nordic Journal of Philosophical Logic (2005)
11. Castelfranchi, C., Dignum, F., Jonker, C.M., Treur, J.: Deliberative normative agents: Principles and architecture. In: 6th Int. Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL'99). LNCS, vol. 1757, pp. 364–378. Springer (2000)
12. Castelfranchi, C., Falcone, R.: Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In: Demazeau, Y. (ed.) ICMAS. pp. 72–79. IEEE Computer Society (1998)
13. Clavel, M., Eker, S., Lincloln, P., Meseguer, J.: Principles of Maude. In: Proc. First Int. Workshop on REwriting Logic and its Applications. ENTC, vol. 4, pp. 65–89. Elsevier (1996)
14. Corkill, D., Durfee, E., Lesser, V., Zafar, H., Zhang, C.: Organizationally Adept Agents. In: 12th International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN@AAMAS 2011). pp. 15–30 (2011)
15. Coutinho, L.R., Sichman, J.S., Boissier, O.: Handbook of Research on Multi-Agent Systems : Semantics and Dynamics of Organizational Models, chap. Modelling Dimensions for Agent Organizations, pp. 18–50. Information Science Reference Publisher (2009)
16. Criado, N., Argente, E., Noriega, P., Botti, V.J.: Towards a normative bdi architecture for norm compliance. In: Proceedings of 11th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN at MALLOW2010) (2010)
17. Dastani, M., Dignum, V., Dignum, F.: Role-assignment in open agent societies. In: Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS'03). ACM Press, Melbourne (2003)
18. Dastani, M., Riemsdijk, M.B.V., Hulstijn, J.: Enacting and deacting roles in agent programming. In: In Proceedings of the 5th International Workshop on Agent-Oriented Software Engineering (AOSE). vol. 189-204, p. 3382. Springer (2004)
19. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis, Universiteit Utrecht (2004)
20. Dignum, F., Dignum, V., Thangarajah, J., Padgham, L., Winikoff, M.: Open agent systems??? In: Proceedings of the 8th International Workshop on Agent-Oriented Software Engineering (AOSE'07). LNCS, vol. 4951, pp. 73–87. Springer (2008)
21. Hannoun, M., Sichman, J.S., Boissier, O., Sayettat, C.: Dependence relation between roles in a multi-agent system: towards the detection of inconsistencies in organization. In: Sichman, J.S., Conte, R., Gilbert, N. (eds.) Proceedings of the First Int. Workshop on Multi-Agent Systems and Agent-Based Simulation. pp. 169–182. LNAI 1534, Springer, Berlin (1998)
22. Hubner, J.F.: Um Modelo de Reorganizacao de Sistemas Multiagentes. Ph.D. thesis, Universidade de Sao Paulo, Escola Politecnica (2003)
23. Hübner, J.F., Boissier, O., Sichman, J.S.: Using a multi-agent organization description language to describe contract dynamics in virtual enterprises. In: IAT. pp. 672–678 (2005)
24. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents. Journal of Autonomous Agents and Multi-Agent Systems (2009)

25. Hübner, J.F., Sichman, J.S., Boissier, O.: Developing Organised Multi-Agent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels. Agent-Oriented Software Engineering 1(3/4), 370–395 (2007)
26. Kollingbaum, M.J., Norman, T.J.: Noa - a normative agent architecture. In: Gottlob, G., Walsh, T. (eds.) IJCAI. pp. 1465–1466. Morgan Kaufmann (2003)
27. Meneguzzi, F., Luck, M.: Norm-based behaviour modification in BDI agents. In: Proceedings of the eighth international joint conference on autonomous agents and multiagent systems (AAMAS'09). pp. 177–184. Budapest (2009)
28. Meneguzzi, F., Oren, N., Vasconcelos, W.: Using constraints for norm-aware BDI agents. In: The Fourth Annual Conference of the International Technology Alliance. London, UK (2010)
29. Panagiotidi, S., Vázquez-Salceda, J.: Norm-aware planning: Semantics and implementation. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03. pp. 33–36. IEEE (2011)
30. van Riemsdijk, M.B., Dignum, V., Jonker, C.M., Aldewereld, H.: Programming role enactment through reflection. In: 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'11). vol. 2, pp. 133–140. IEEE (2011)
31. van Riemsdijk, M.B., Hindriks, K.V., Jonker, C.M.: Programming organization-aware agents: A research agenda. In: Proceedings of the Tenth International Workshop on Engineering Societies in the Agents' World (ESAW'09). LNAI, vol. 5881, pp. 98–112. Springer (2009)
32. Sabater, J.: Evaluating the regret system. Applied Artificial Intelligence 18(9-10), 797–813 (2004)
33. Sichman, J.S., Conte, R., Castelfranchi, C., Demazeau, Y.: A social reasoning mechanism based on dependence networks. In: ECAI. pp. 188–192 (1994)
34. Vazquez-Salceda, J.: The role of Norms and Electronic Institutions in Multi-Agent Systems. Whitestein Series in Software Agent Technology, Birkh-user Verlag AG, Switzerland (2004)

Chapter 4
Adaptive Agent Organisations

Estefania Argente and Holger Billhardt and Carlos Cuesta and Sergio Esparcia and Jana Goermer and Ramon Hermoso and Kristi Kirikal and Marin Lujak and J.S. Perez-Sotelo and Kuldar Taveter

Abstract The capability of autonomously adapting to changing conditions is a feature that requires agents to be able to alter their own configuration and even their own composition. Adaptive agent organisations should consider agent self-

Estefania Argente
Departamento Sistemas Informáticos y Computación, Universitat Politècnica de Valencia, Valencia, Spain e-mail: `eargente@dsic.upv.es`

Holger Billardt
Centre for Intelligent Information Technology (CETINIA), University Rey Juan Carlos, Madrid, Spain e-mail: `holger.billhardt@urjc.es`

Carlos E. Cuesta
School of Computer Science and Engineering, Rey Juan Carlos University, Madrid, Spain e-mail: `carlos.cuesta@urjc.es`

Sergio Esparcia
Departamento Sistemas Informáticos y Computación, Universitat Politècnica de Valencia, Valencia, Spain e-mail: `serparcia@dsic.upv.es`

Jana Görmer
Department of Informatics, TU Clausthal, Clausthal, Germany e-mail: `jana.goermer@tu-clausthal.de`

Ramon Hermoso
Centre for Intelligent Information Technology (CETINIA), University Rey Juan Carlos, Madrid, Spain e-mail: `ramon.hermoso@urjc.es`

Kristi Kirikal
Tallinn University of Technology, Tallinn, Estonia e-mail: `kristi.kirikal@gmail.com`

Marin Lujak
University Rey Juan Carlos, Madrid, Spain e-mail: `lujak@ia.urjc.es`

J. Santiago Perez-Sotelo
CETINIA - GIA, University Rey Juan Carlos, Madrid, Spain e-mail: `josesantiago.perez@urjc.es`

Kuldar Taveter
Tallinn University of Technology, Tallinn, Estonia e-mail: `kuldar.taveter@ttu.ee`

adaptivity and reorganisation processes. In this chapter, a review of methods for designing and/or implementing adaptive agent organisations will be given.

4.1 Introduction

It is well known that the growing complexity of software is emphasizing the need for systems that have autonomy, robustness and adaptability among their most important features. Nowadays it is also accepted that MAS have been developed in artificial intelligence area as a generic approach to solve complex problems. However, in order to fulfil their promise of generality and extensibility, they should also reach self-adaptivity, i.e. the capability of autonomous adaption to changing conditions. This feature requires agents to be able to alter their own configuration, and even their own composition and typing. Therefore, their reorganisation can be seen as the first necessary steps to reach actual self-adaptation.

In this chapter, first we present some basic concepts about agent adaption that have been broadly used. Next, in section 4.3, we present an approach to deal with adaptation in Virtual Organisations, in which we propose several guidelines for identifying internal and external forces that motivate organisational change, studied in depth in Organisation Theory [23]. Thus, in subsection 4.3 we describe how to define an Adaptive Virtual Organisation using an Organisational Theory approach.

In section 4.4, we detail a framework for Adaptive Agent Organisation that provides an architectural solution to tackle the dynamism of organisations. This framework implies an evolving architectural structure based on combining predefined controls and protocols, handled in the context of a service-oriented, agent-based and organisation-centric framework.

As explained in Chap. 1 Sect. 1.4, software-intensive systems can be seen as sociotechnical systems that consist of interacting agents. The methods for designing adaptive sociotechnical systems can be borrowed from social sciences. In section 4.5, we analyse the differences between social and technical systems and we introduce requirements which should be considered while designing sociotechnical systems. A case study of adaptive and iterative development is then introduced and explained in section 4.6.

A particularly difficult task for an agent is deciding with whom to interact when participating inside an Open Multi-Agent System. In section 4.7 we present a mechanism that enables agents to take more informed decisions regarding their partner selection. This mechanism monitors the interactions in the Open Multi-Agent System, evolves role taxonomy and assigns agents to roles based on their observed performance in different types of interactions. So then this information can be used by agents to estimate better the expected behaviour of potential counterparts in future interactions.

Dealing with groups of autonomous agents the IT-ecosystem can balance on one hand its adaptability and on the other hand its controllability. In section 4.8 we present group-oriented coordination, in which we explain how this kind of cooperation and coordination mechanisms finds an equilibrium for global and individual objectives. We apply the group oriented coordination on a simple example allowing agents to form faster and slower groups.

Finally, we also consider the problem of coordinating multiple mobile agents which collaborate to achieve a common goal in an environment with variable communication constraints. In section 4.9 we present a task assignment model for cooperative MAS, in which a team of mobile agents has to accomplish a certain mission under different inter-agent communication conditions.

4.2 Concepts on Adaptive Agent Organisations

Adaptive organisations is a key research topic inside the MAS domain. In this section we will present and discuss relevant concepts and definitions for adaptive agent organisations, mainly focusing on adaptive Organisation-Centred MAS (OCMAS).

Aldewereld et al. [1] define adaptive software systems as 'those that must have the ability to cope with changes of stakeholders' needs, changes in the operational environment, and resource variability'. DeLoach et al. [13] define adaptive organisations as distributed systems that can autonomously adapt to their environment. The system must be provided with organisational knowledge, by which it can specify its own organisation, based on the current goals and its current capabilities.

Picard et al. [41] describe that an OCMAS is adaptive when it changes whenever its organisation is not adequate, i.e. the social purpose is not being achieved and/or its structure is not adapted to the environment. This situation occurs when the environment or the MAS purposes have changed, the performance requirements are not satisfied, the agents are not capable of playing their roles in a suitable way or a new task arrives and the current organisation cannot face it. In this case, adaptation implies modifying both organisation specification (modifying tasks, goals, structure) and role allocation.

Dignum and Dignum [15] state that in order to keep effective, organisations must maintain a good fit with the environment. Changes in the environment lead to alterations on the effectiveness of the organisation and therefore in a need to reorganize, or at least, the need to consider the consequences of the change to the organisation's effectiveness and, possibly, efficiency. On the other hand, organisations are active entities, capable not only of adapting to the environment but also of changing that environment.

Summarizing, an Adaptive Organisation in MAS presents the following properties:

- The organisation changes if its environment forces it to do so.
- Changes in goals, internal requirements, etc. of the organisation could also force a change.
- The organisation is considered to be an open system since the environment might change and external agents may join the organisation.

- The organisation is populated by agents playing different roles, some of them being responsible of deciding about change.

Based on these previous works, a definition for *Adaptive Virtual Organisation* is proposed in [19]: An Adaptive Virtual Organisation is a virtual organisation that is able to modify both its structural (topology, norms, roles, etc.) and functional (services, tasks, objectives, etc.) dimensions in order to respond or to be ahead of changes produced in its environment, or by internal requirements, i.e. if it detects that its organisational goals are not being achieved in a satisfactory way.

When executing an adaptation process in an OCMAS, two types of change can be distinguished: dynamical (behavioural) and structural [16]. *Dynamical changes* are those in which the structure of the system remains fixed, while agents and aspects like role enactment are modified. *Structural changes* are produced in structural elements of the system, like roles, topology or norms.

Regarding dynamical changes, there are three types that must be considered:

- A new agent joins the system. It is necessary to reach an agreement to join the organisation, playing a particular role that indicates the rights and duties of the agent that plays that role.
- An agent leaves the system. It is necessary to determine if this operation is possible, taking into account certain imposed conditions by the MAS management. Sometimes, it could not be appropriate to allow an agent playing a specific role to leave the system. In other moments, it may be convenient to reassign that role as soon as the agent leaves the system.
- Instantiation of the interaction pattern. A change of this kind consists of two agents that carry out a certain interaction pattern and reach an agreement to follow a protocol adjusted to this interaction pattern. In this kind of changes there are included, for example, changes related to the role enactment process, changes in the agents that are providing a service or in the set of active norms, etc. These changes force agents to modify their interaction pattern.

Regarding structural changes, there are two ways to carry out a structural change in an organisation:

- Self-organisation: implies the emergency of changes, appeared because of the interaction between agents in a local level, that generates global level changes in the organisation.
- Reorganisation: designed societies are adapted to modifications in the environment by adding, deleting or modifying their structural elements (roles, dependencies, norms, ontologies, communication primitives, etc.).

Self-organisation changes are bottom-up, where an adaptation in the individual behaviour of the agents will lead to a change in the organisation in an emergent way. Thus, self-organisation is an endogenous process (carried out by the agents). Agents are not aware of the organisation as a whole, they only work with local-level information to adapt the system to environmental pressures by indirectly modifying the organisation. Therefore, agents, using local interactions and propagation, modify the configuration of the system (topology, neighbours, influences, differentiation).

There are some proposals about MAS self-adaptation, and here we present some of them as an example. Gardelli et al. [22] use artifacts as a tool to introduce self-organisation inside a MAS. In the work by Kota et al. [32] a pair of agents estimate the utility of changing their relation and take the appropriate action accordingly. ADELFE [9] is a methodology that proposes the design of agents that are able to modify their interactions in an autonomous and local way in order to react to the changes that are produced in their environment. MACODO [26] is a middleware that offers the life-cycle management of dynamic organisations as a reusable service separated from the agents.

Regarding reorganisation, it is a top-down approach, so that a modification in an organisational aspect will produce changes in agents composing the organisation. Reorganisation can be both an endogenous or an exogenous process (controlled by the user or by an external system), referred to systems where the organisation is explicitly modified through specifications, restrictions or other methods, in order to ensure a suitable global behaviour when the organisation is not appropriate. Agents are aware of the state of the organisation and its structure, being able to manipulate primitives to modify their social environment. This process can be initialized by an external entity or by the agents, directly reasoning over the organisation (roles, organisational specification), and the cooperation patterns (dependencies, commitments, powers).

The OCMAS community of researches has presented different proposals to deal with adaptive organisations, one of each using their own point of view. Three of these works (the ones from ALIVE [1], Dignum and Dignum [18], and Hoogendorn [28]) state that they based their knowledge about organisational change on the human Organisation Theory. Also, both human and agent organisations have many elements in common. These three proposals conceive organisational change as an endogenous process, where agents populating the organisation will be responsible for organisational adaptation. These agents could be all the agents populating the organisation, or just only a set of agents (typically playing a management role) that are organisation aware, and are provided with all the knowledge they need to understand modifications and to perform changes inside the organisation.

Nevertheless, the approach followed by MOISE [29] is different. In this case, MOISE was not initially conceived to give support to adaptation, but it was later adapted to provide support to reorganisation. Roles inside MOISE are distributed in different groups, so as to give support to adaptation, a new group, external to the organisation, was added. This makes the process of change to be exogenous, making a difference with respect to the rest of proposals. However, this process still preserves the common steps for reorganisation, including monitoring, design and implementation of change.

It must be noticed that these proposals follow a formal approach to define change. Dignum and Dignum have an interesting background in formal and logic languages, with proposals like OperA [14] or LAO [17]. ALIVE also takes inspiration from previous proposals by Dignum and Dignum, since it is a joint project of some European universities, including the Universities from Delft and Utrech, where Dignum and Dignum develop their work. Therefore, their proposals are very similar. Hoogendoorn also works with a formal logic language, TTL, that makes easier to check the correctness of the definition of a system and its adaptation process.

The next sections of this chapter present proposals for designing and developing adaptive agent organisations and other related elements. These proposals are mixed, since some of them follow a reorganisation, top-down approach to define organisations, and some others define a self-organisation, bottom-up development.

4.3 Adaptive Virtual Organisations using an Organisational Theory approach

As presented in Chap. 1 Sect. 1.3 of this book, the Virtual Organisation concept is based on human organisations. Therefore, changing factors in a human organisation can also be considered as changing factors in a Virtual Organisation. In the domain of the Organisation Theory [23] these factors are known as *forces* that lead to organisational change. Those forces can be *internal* or *external*, depending on where their source is located. Usually, a change in the environment is the main external cause, while a change in the requirements or goals of the organisation is the most common internal reason for change. Obviously, these changes are generic, and specific changing factors must be defined depending on the domain of each system.

In the following, we present the most common forces, both internal and external, and we also depict our proposal for dealing with these forces, thus turning a Virtual Organisation into an Adaptive Virtual Organisation [19].

Forces that drive organisational change. An organisational change is produced by one or some forces that can be differentiated by their nature. Some organisations are more vulnerable than others due to the pressure of change, such as organisations with diffuse objectives, uncertain support, unstable values and those that face a declining market for their products and services.

The ***external forces*** are those that promote changes inside an organisation due to changes in its environment. Thus, the external forces are referred to the environment where the organisation is located. They are due to elements such as other organisations that populate the same environment (and some of them suppose competence) or different heterogeneous agents in the same environment. Among external forces, the following forces can be found: (a) *Obtaining resources*: if a failure occurs in an organisation while obtaining resources, it leads to an organisational change to guarantee organisational survival [2]. Therefore, it could be necessary for organisational survival to improve the way in which resources are obtained; (b) *Market forces*: Requirements of products and services of an organisation by internal and external agents may change through time, so the number of requests for a product or a service that an organisation is offering is not constant. Therefore, organisations that offer services or products that nobody is requiring have no reason to exist, so they will disappear if they do not decide to change in order to offer new products and services that are currently being demanded [3]; (c) *Generalisation*: some organisations that are unable to acquire enough resources by specializing themselves in a limited range of products or services manage to survive by becoming generalists, i.e. by offering a set of products and services that are oriented to a more general purpose, thus increasing their number of potential customers; (d) *Decay and deterioration*: An organisation can be affected by environmental changes that will make its objectives obsolete or they could lose their sense [3]; (e) *Technological changes*: An organisation can adopt new technology in order to improve its productivity inside the market where it is developing its activities [8]; (f) *Competence*: One of the reasons for the organisational change is the existence of organisations with a similar purpose, turning into competence for them [8]; (g) *Demographical features*: Since organisations are open systems, agents populating them and their environment are heterogeneous. An organisation must control this diversity in an effective way, paying attention to the different needs of these agents, but trying to avoid malicious and/or self-interested behaviours by them [36]; (h) *Laws and regulations*: There can be external laws that could affect the environment of an organisation or its neighbours organisations [8]; and (i) *Globalisation*: Globalisation refers to the increasing unification of the world's economic order through reduction of such barriers to international trade as tariffs, export fees, and import quotas [46]. The goal is to increase material wealth, goods, and services through an international division of labour by efficiencies catalysed by international relations, specialisation and competition.

The ***internal forces*** of an organisation are signals produced inside an organisation, indicating that a change is necessary. Thus, it is important to clearly define these forces, in order to monitor them and to achieve the change in the most appropriate form and moment. The internal forces are: (a) *Growth*: When an organisation grows in both members or budget, it is necessary to change its structure to a more hierarchical organisation, with higher levels of bureaucratisation and differentiation among its members [3]; (b) *Power and political factors*: The most powerful members of an organisation may have different objectives than agents in a lower hierarchical level, which can be even different from the organisational objectives. The organisation may assure (for instance, by means of observers) that manager agents do not impose their objectives above organisational objectives [3]; (c) *Goal succession*: There are certain organisations that disappear after reaching their goals. However, some other organisations look for new goals to achieve. Therefore, these organisations will continue with their existence; (d) *Life-cycle*: Some existing organisations follow the classic life-cycle model. Thus, they appear, grow, change, and disappear, to give way to other organisations [8]; (e) *Human resources*: Managers of the organisation must control that their agents are committed with the organisation, present an adequate behaviour and their performance is acceptable; (f) *Decisions and managers behaviour*: Industrial disputes between agents and their supervisors inside organisations are an important force for change. If a subordinated agent disagrees with his/her supervisor, he/she could ask for new tasks to develop inside the organisation. If the management approves his/her petition, an action must be carried out; (g) *Economical restrictions*: Organisations want to maximize their performance. Therefore, they will try to obtain maximum benefits using the less possible amount of resources. If it is considered that too much resources are being consumed, a change can be necessary; (h) *Merging and acquisitions of organisations*: One of the internal forces that will drive the organisational change is the merging of two or more organisations, or the acquisition of one organisation by another, leading to bigger organisations where their structure and members should be reorganized. Merging will allow to compete from a better position with other organisations; and (i) *Crisis*: If an organisation is in a crisis due to a sudden drop of its efficiency, a possible solution is a deep organisational change, modifying structural and/or functional elements, depending on the specific needs of the organisation.

How to identify an acting force. A key issue when dealing with adaptation is that forces that drive organisational change should be correctly detected. We have defined a guideline (Table 4.1) [19] for detecting when a force is acting over the organisation. For each common force that leads to organisational change, a guideline has been completed. On each of these guidelines, there are represented the different factors that should be monitored in order to detect that a force is acting. It must be noted that not all factors are required to be detected in order to state that a specific force is acting over an organisation, but just a subset of these factors could be able to trigger a force. It is possible for each factor to come from different sources, such as from the behaviour of an agent, or the level of fulfilment of set of goals.

Guideline for detecting a driving force	
Field	**Description**
Name	Name of the force which is able to be detected by following this guideline
Description	Describes how this force acts over an organisation
Type	*Internal* or *external*, depending on whether this force comes from the own organisation or its environment
Factors	
Name	The name of the factor that helps identifying the force
Description	The description of this factor
Type	The type of the factor (e.g. behaviour of agent/role, goal achievement, etc.)
Value	The value that this element must reach/not reach in order to be considered as a factor for change
Triggers	Specifies whether this factor triggers the force by itself, of other factors are required in order for a force to start acting over an organisation

Table 4.1 Guideline for detecting a force that drives organisational change

Solution for preventing damage or taking advantage from a force. We have also defined a guideline (Table 4.2) [19] for identifying the different organisational actions that should be carried out in the organisation in order to take advantage or to prevent damage from a specific force.

Each solution is described by its name, its description, the force (or forces) that are intended to take advantage of or trying to reduce its damaging effects over the organisation. Also, this guideline points out the factors for detecting a force that must appear along with the force in order to be possible to apply this solution, as well as the specific roles that will carry out this solution.

The organisational actions are those actions that will produce a change in the organisational definition when they are executed. Taking the Virtual Organisation Formalisation (built by the *OS* referring to the Organisational Specification, *OE* to the Organisational Entity, and ϕ to the Organisational Dynamics, as explained

Solution for preventing damage or taking advantage of a force	
Field	**Description**
Name	Name of the solution
Description	Text describing this solution
Force	The force that must be acting to apply this solution
Factor	The set of factors that must be detected in order to be able to apply this solution
Actions	The set of actions that must be carried out to apply this solution
Roles	The responsible roles for applying this solution

Table 4.2 Guideline for applying a solution

in Chap. 1 Sect. 1.3) as reference, the execution of an organisational action *oa* in a virtual organisation vo_i implies that the time increases ($t \rightarrow t+1$). An organisational action is defined as:

$$\frac{vo_i \quad \rightarrow_{oa} \quad vo_i'}{\langle OS, OE, \phi \rangle \rightarrow \langle OS', OE', \phi' \rangle} \tag{4.1}$$

This expression states that a virtual organisation vo_i, at a given time t, carries out an organisational action *oa* that causes a change in the organisational state, being vo_i' the new state of the organisation, at a time t+1. Notice that it is not mandatory for an organisation to change every component in order to change its state, i.e. ($OS = OS' \vee OE = OE' \vee \phi = \phi'$).

The two proposed guidelines have been applied, as an example, to the description of the external force ”Obtaining resources” (Table 4.2), which is explained as follows:

Obtaining resources (External force). Resources are commonly used as raw materials to produce the results of the services of an organisation. Therefore, if a service is called, and it has a precondition that specifies that a resource is needed to execute a service, but the resource cannot be obtained using the current organisational structure, it is necessary to look for a solution. In this case, the most appropriate solution could be to move any of the entities to a workspace where this resource is available (i.e. place the entity inside the population of this workspace).

Detecting ”Obtaining resources” external force	
Field	**Description**
Name	Obtaining resource
Description	A resource is not able to be accessed by an organisation
Type	External
Factors	
Name	Successful calls to a service
Description	If the rate for successfully executing a service is lower than a given threshold, it means that this force is acting
Type	Service providing rate
Value	Threshold
Triggers	This factor itself triggers the force

Table 4.3 Example of the guideline for detecting a force that drives organisational change

The solution to this force (Table 4.4) is to move an entity of the organisation to a workspace where this resource is available. In our approach, that means to execute the organisational action 'move entity' to a workspace of the organisation $vo_1 \in \mathscr{V}\mathscr{O}$ at a given time t.

Solution for "Obtaining resources" external force	
Field	**Description**
Name	Move entity to a workspace
Description	An entity of the organisation is placed in a workspace where the artifact is located
Force	Obtaining resource
Factor	Threshold of successfully executing a service
Actions	Move entity to a workspace
Roles	The responsible roles for applying this solution

Table 4.4 Example of the guideline for applying a solution

A different solution is to negotiate with another organisation, in order to be able to go inside this organisation to get resources or to allow an external agent which is able to get this resources to join the organisation. Notice that this solution is appropriate just in case the organisation is not able to find the required resource among its perceived workspaces. So, it must look for it outside the organisation.

4.4 A Framework for Adaptive Agent Organisations

It is well known that the growing complexity of software is emphasizing the need for systems that have autonomy, robustness and adaptability among their most important features. It is also accepted nowadays that MAS have been developed in artificial intelligence area as a generic approach to solve complex problems. However, in order to fulfil their promise of generality and extensibility, they should also reach self-adaptivity, i.e. the capability of autonomously adapting to changing conditions. This feature requires them to be able to alter their own configuration, and even their own composition and type. Their reorganisations can be seen, therefore, as the first necessary steps to reach actual self-adaptivity.

This section proposes an architectural solution to tackle the dynamism, which will be supported by an emergent agreement - an evolving architectural structure based on combining predefined controls and protocols. These are handled in the context of a service-oriented, agent-based and organisation-centric framework [40]. Next, we will discuss not only the architectural framework but also the mechanisms to change their composition patterns and element types, which are necessary to achieve real self-adaptivity.

The Basic Framework for Adaptive Organisations. As the proposed approach is based on service-oriented concepts, the main idea is to export the agent system as a system of services, and the environment must be truly adaptive and dynamic, it requires the use of rich semantic and highly technological capabilities. Therefore, it is considered a wise use of *agents* in a broader context, with an upper layer of services added to provide, in particular, the interoperability feature. It is easy to conceive a service to present the operational capabilities of an agent or, even better, of a collection of agents as an organisation, which in turn provides services. Using agents allows the explicit treatment of semantics, a structured coordination, the use of a methodology to service development, to structure them into organisations, and the use of their learning capacity, among others features.

Implicit in the definition of MAS is the need to *register* agents in the system, to separate those ones who belong to the architecture from those who do not. The same approach will be used to identify services. To allow their external access, they will be explicitly registered and grouped as part of a service.

The current research, which is included as part of the OVAMAH project [39], is extending the objectives of the original platform THOMAS [6]. Besides providing the necessary technology for the development of virtual organisations in open environments, it will allow to facilitate dynamic answers for changing situations by means of the adaptation and/or evolution of the organisations. For example, agents forming an organisational unit could create (or remove) another unit, affecting the groups of the system; decide the moment to add or delete norms; the social relationship between roles could change at runtime, the conditions to activate/deactivate, as well as the cardinality of roles; the system topology (given by the relationships) could be changed also at runtime and then validate the changes with objectives and organisational type; the services could be matched to new roles; etc.

The framework is evolving (currently adapting to OSGi [38] specification) and the applications are modularizing into smaller entities called bundles. These entities can be installed, updated or removed on the fly and dynamically, provide the ability to change the system behaviour without ever having to disrupt its operation. Among the services provided by this standard, the Service Tracker appears as particularly relevant, in the light of the proposed approach. This service makes possible to track other registered services on the platform. It is used to ensure that the services to be provided are still available or not.

In summary, the evolution of the agreement-based approach, including the concepts and constructs that it describes, has already shown its relevance. The main concern now, beyond performance issues, is the essential dynamism and the adaptive functionality required by the underlying architecture.

Adaptive Organisations based on *Initiatives*. A group of individuals can be arranged into certain structures, depending on concrete goals, and they can be formed by using two different kinds of mechanisms: *controls* and *protocols*, which are both based on limiting the range of available actions. The formers can be seen as elements that either enforce or forbid specific interactions (or architectural connections). Self-adaptive structures, being typically centralized [5], show many classic examples of this kind: most of them manifest explicit control loops, inspired in regulators of classic control theory. On the other hand, protocols, which either enable or channel behaviour, are based on consensus and agreements. They can be described generically as the way to control decentralized (even distributed) structures [20]. Basically, when protocols are present, every agent knows the way to interact with the rest; it is necessary to comply with them to be able to communicate, but at the same time they are also regulating the development of the interacting structure itself.

These two mechanisms define a wide spectrum of regulation, in which agent organisations and their architectures are simultaneously harnessed by atomic, unary controls (norms, limits, locks, control loops or constraints) and multiple, connective protocols (hubs, bridges, channels, or spaces). It is important to note that the purpose of these mechanisms is to ”discover” a suitable structure of controls and protocols so that a global structure can emerge. These elements make possible to define the main inner structures in order to obtain agreement-based organisations. Once a primary structure can be defined, an elemental group emerges as a preliminary organisation, which will be referred as an *initiative*: not yet fully established, but still evolving.

Nevertheless, the *initiative* can continue growing and mutating because of its adaptive nature, but when it has some ”stable” structure, it can be called organisation. This ”stable” structure is achieved when all the participants can afford the necessary agreement in order to gain the objective. This process can be thought as the system moving to a new state, in which the structure of the ”past” is supplanted by a ”new” emergent structure. Obviously, this novel structure admits new elements because of the dynamic environment, but now one of its goals is to reinforce its nature.

An *initiative* can be generated from patterns, named *adaptation patterns*, where the term is used in an architectural sense. They are pre-designed from the required services of an *initiative* and the corresponding semantic refining. Some of them have been already identified, and receive such names as Façade, Mediator, or Surveyor, among others (see Figure 4.1). The patterns represent a fragment of a static structure, leading to a dynamic one, the *initiative*, reaching a ”stable” form, the organisation.

Adaptation Patterns. As already noted, the adaptation patterns are pre-designed from the required services of an *initiative* and for the corresponding semantic refinement. Particularly, these are not classic object-oriented patterns, because they are defined in a different context: they are architectural patterns.

According to [45] it is possible to classify the architectural design patterns as follows: monitoring (M), decision-making (DM), or reconfiguration (R) based on their objective. M and DM patterns can also be classified as either creational (C) or structural (S), as defined in [21]. Likewise, R patterns can also be classified as behavioural (B) and structural (S) since they specify how to physically restructure an architecture. Several of these patterns have been already identified for the proposed approach. In Table 4.1, for instance, three of them are described: Façade, Mediator, and Surveyor.

Obviously, there are more patterns and not all of them describe only roles. For instance, the Surveyor Election defines the protocol (one among many) to decide the next surveyor; and Surveyor Change describes a protocol to demote the current surveyor and forward its knowledge to a new one.

All these pre-figured changes are applied to organisations that have reached a quiescent or safe state for adaptation [33]. In this case, namely pure adaptation, the importance lies in the way that an existing organisation has to adapt to a new

Name	Category	Description
Façade	M, S	To be able to easily interact with an organisation, which still lacks a defined structure, some agent has to represent the organization itself in terms of interaction. This agent redirects any incoming communication.
Mediator	R, B	During the emergence process, the organization is not yet established, and data services are probably not working. Some agent must act as a *mediator*, which makes possible to access to data sources, although indirectly, and also to perform the necessary (*semantic*) translations.
Surveyor	R, S	During the emergence process, at least one agent must monitor the growing of the initiative itself, both to decide when new elements are inserted, and also when the initiative forms a "stable" organization. It has access to the pattern library and decides when a certain pattern must be triggered.

Fig. 4.1 Adaptation Patterns: architectural design patterns

behaviour. First, it has to realize that a change has occurred, i.e. a change can emerge in an intrinsic way [44], and then it has to adapt itself.

There are several scenarios to develop this adaptive behaviour, reaching ultimately a ”stable” configuration for an *initiative* which therefore becomes an organisation. For example, in an emergency situation, some police cars can arrive to the crisis area but no one is the leader of the group. They follow a previous internal protocol to choose a leader (even hierarchy is a protocol), and this agreement generates a preliminary organisation. This is what it is called a *generative protocol*. When the individuals follow this kind of protocols, they define implicit structural patterns.

Lifecycle of Self-Organizing Structures. As we already noted, depending on concrete goals, any group of individuals can be arranged into certain structures by using controls and protocols. These elements will make possible to define the main inner structures in order to obtain agreement-based organisations. Once a primary structure is defined, an ”elemental” group emerges as a preliminary entity: the *initiative*. It will grow with the environmental dynamics until become into a ”stable” organisation.

Figure 4.2 summarizes briefly the lifecycle of our self-organizing structures [12]. This cycle can begin with a single agent, which is able to perform certain interactions and has the potential to export some services. Initially, it does not belong to any organisation when reaches the system. However, it complies to a number of predefined controls and protocols, which ”guide” the agent’s interaction and enable it to maintain structured conversations with others, composing informal groups of agents.

When an external change occurs, the system must react with an adaptive behaviour, and this is the functionality that must trigger the formation of the self-organizing structures (organisations). The system is provided with a number of

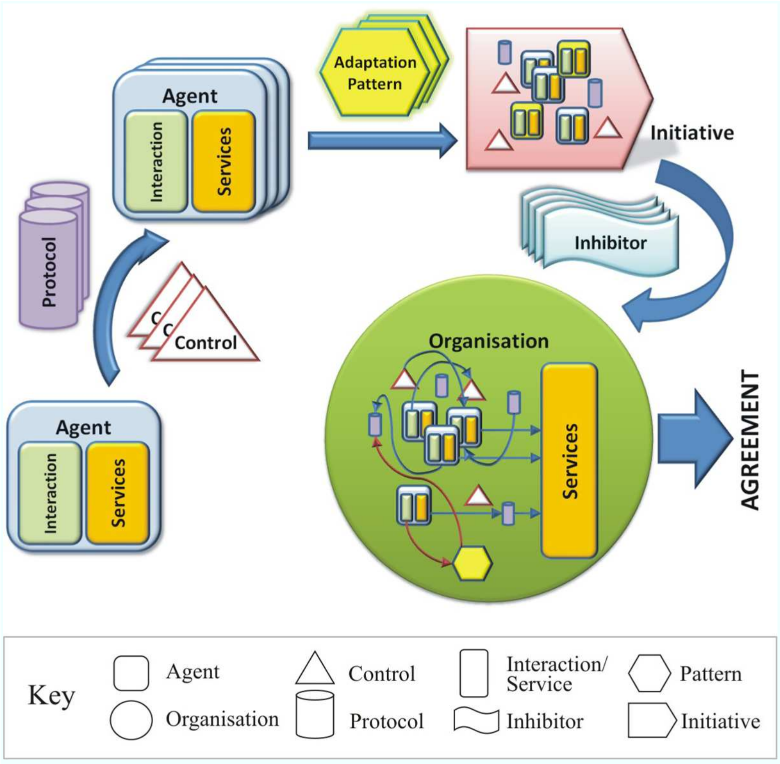


Fig. 4.2 Lifecycle of a self-organizing structure (from [12])

adaptation patterns in order to achieve some desired reaction. These patterns are partial definitions of elements and relationships, which include enough information for an agent to learn how to perform some behaviour. Therefore, under the guidance of an adaptation pattern, certain agents within the group acquire specific functions, and begin to form an actual structure: this is the *initiative*. Of course, these organisations are able to evolve themselves, and to participate in larger agreements [12].

As already noted, the system is ultimately conceived as a service-oriented architecture; so methodologically, the first stable organisations must be considered as the providers for certain high-level services. Then, these services must be proposed as the starting point for the functional definition of those first organisations.

4.5 Adaptive Agent Organisations with Sociotechnical Systems

The challenges in creating software for modern complex and distributed computing environments are described by Sterling & Taveter [47]. They are time-sensitivity, uncertainty, unpredictability, and openness. It is a problem how to design systems that work effectively in the modern environment, where computing is pervasive, people interact with technology existing in a variety of networks, and under a range of policies and constraints imposed by the institutions and social structures that we live in. The key concepts that Sterling & Taveter [47] use for designing open, adaptive, distributed, and self-managing systems are *agents* and *sociotechnical systems*. An *agent* is suitable as a central modelling abstraction for representing distributed interconnected nodes of the modern world. A *sociotechnical system* encompasses a combination of people and computers, hardware and software.

The novelty of the approach to be presented in this section is that it shows how treating software-intensive systems as sociotechnical systems that consist of interacting agents facilitates the design of such systems. We claim that the methods for designing adaptive sociotechnical systems should be borrowed from social sciences rather than from exact sciences. We show how it can be done.

To start with, it is crucial to understand how social and technical systems differentiate each other. Only when this understanding is achieved, it will become possible to form the foundations for designing systems. On this grounding, in this section we first analyse differences between social and technical systems. Then we introduce requirements which should be considered while designing sociotechnical systems. Finally a case study of adaptive and iterative development will be introduced and explained.

Social Systems. Sociotechnical systems are more complex than merely technical systems. Methods of exact sciences are not applicable to social systems. As Prigogine [43] pointed out, the world is a complex system which develops in irreversible time. It is impossible to re-create the same situation in a social environment because social experiments are not conducted in a laboratory. Social experiments have impact on society and therefore initial conditions will also change. A social system can be viewed as having two kinds of statuses. These modes are ”is” and ”is not” or ”agree” and ”disagree”, depending on the situation. The action of choosing a status by an agent triggers some event.

Popper states that no scientific predictor - whether a human scientist or a calculating machine - can possibly predict, by scientific methods, its own future [42]. Luhmann [34] (p.177) claims that establishing and maintaining the difference between system and environment becomes the problem, because for each system the environment is more complex than the system itself. Allert and Richter [4] lead this thought to the conclusion by saying that ”the difference system/environment is not ontological but an epistemological - it is continuously constructed by the observer, based on his actual motive”.

Technical Systems. In contrast to social systems, technical systems can be studied by applying the methods of exact sciences. The experiments conducted with technical systems in a laboratory are repeatable and the same outcome is expected from them. For example, the results from chemical experiments should be identical when the same experiment is repeated under the same initial conditions.

Also in software development time does not have an effect on the system when the system is not intentionally changed. This means that while testing the system, the same test case should end with the same results. A software system is considered to be of a high quality when it functions as expected.

In technical systems, the difference between system and environment is drawn from early on. For example, use cases of UML pressure the modeller to decide the system boundary already at the beginning of requirements engineering. We can conclude this section by stating that important features of technical systems are predictability and clear system/environment difference.

Considerations for Designing Sociotechnical Multi-Agent Systems. In the previous section we pointed out that social systems are essentially different from technical systems. Here, we supplement this reasoning by elaborating considerations for designing sociotechnical systems. The most important point that we argue is that *sociotechnical systems should be designed by following principles of behavioural and social sciences*. The rationale for this is that behavioural and social sciences are more complex and therefore their characteristics should be used when designing sociotechnical systems. Only then can social and technical systems be merged.

First of all, we need proper abstractions for engineering sociotechnical systems. Central among such abstractions is that of an *agent*, which we term as an active entity, such as person, software agent, or robot. It is worthwhile to point out here that also people are agents. People live and act in the world and interact with each other. Viewing people as agents helps to conceptualize systems in terms of agents.

Because of continuous time, sociotechnical systems are *here-and-now* systems or run-time systems where agents should adapt to changes in the environment gradually at run-time. It is important that no agent in the system needs to know everything. It is sufficient when an agent knows enough to achieve his/her own goals. If the information required is not available in the current situation, the agent will use the information that is available or will try again after a while. However, that will be another *here-and-now* situation.

Sociotechnical systems should be gradually extendable. In other words, sociotechnical systems do not require that all of their constituent agents should be implemented at once. For example, a human agent can create a one-person company. When one person cannot any more manage with all of the tasks and there are enough resources to hire another person, the organisation can be extended.

In order to develop good-quality sociotechnical systems, the goals of the *system* should be known, usually by human agents. Designing a self-organizing agent system without a known outcome is of no value because the outcome can be order as well as chaos. What is not predictable is how exactly the goals of the system are achieved. As time is irreversible, normally there are several options for achieving the same goal. Therefore adaptive and flexible systems which can keep up with changes occurring at run-time should be designed.

In sociotechnical systems, storing history is not the main priority. Rather it is important for each agent of the system to know from where to obtain information and how to utilize it. More important than having a lot of data in the system is having agents that can interpret data for realizing their goals.

Sociotechnical systems should follow patterns of social systems. In a society, each person is an autonomous agent. No one knows information about the society as a whole but everyone knows the information necessary to fulfil its objectives. Moreover, no one, not even the President, knows all the information about a reasonably large organisation such as Tallin University of Technology. However, society as a complex system works reasonably well, despite the fact that each member of a society knows only a very small part of the whole system. We are convinced that this approach is also applicable to engineering sociotechnical systems.

Based on the preceding arguments, when designing sociotechnical systems, there is no need to describe their environments as accurately as possible (and as was indicated in the *Social Systems* paragraph, it is not even possible). What matters is that each constituent agent of the system knows enough about its specific objectives and about the means of achieving them.

System design for complex sociotechnical systems requires new approaches. In next Section 4.6 we propose a solution for adaptive development of flexible sociotechnical systems in such a way that an environment does not have to be analysed in its entire complexity and the system can be developed adaptively and iteratively to match a continuously changing world.

4.6 Adaptive and Iterative Development

In this section we propose an approach for iterative bottom-up development of sociotechnical systems based on agent-oriented modelling. We claim that this approach is applicable to the systems consisting of human and/or man-made agents. Another claim is that if sociotechnical systems are developed this way, they can be easily adapted to the changing conditions.

The rationale for this approach is that contemporary complex systems can have no agents who know all the information. It is not even necessary to have such agents. Instead, it is important for each agent to know its objectives and the means of achieving them. In our view, sociotechnical systems should be developed in iterative phases. A system in its any phase should include at least one agent who is aware of the goal which should be achieved by the system as a whole, i.e. the system's purpose. As we do not yet live in the world described by Isaac Asimov [7], where man-made agents can create themselves, at the beginning of adaptive development the agent who is aware of the system goal is a human agent. When an agent knows the system goal, a complex system environment does not have to be described in detail. Each agent knows its objectives and this is sufficient to achieve the system's goal.

Our approach is rooted in agent-oriented modelling proposed by Sterling & Taveter [47], which was overviewed in Chap. 1 Sect. 1.4. However, instead of using agent-oriented modelling for just *top-down* development of sociotechnical systems, as proposed by Sterling & Taveter [47], we propose to apply agent-oriented modelling also to iterative *bottom-up* development of sociotechnical systems. We have chosen agent-oriented modelling because it supports well the openness of sociotechnical systems by postponing deciding the system/environment boundary until platform-independent design.

In our approach, agents belong to different abstraction levels. First-level agents are always used because they do the actual work, such as assembling cars or cell phones on a production line. Therefore agents of the first-level are created before agents of other levels. Figure 4.3 represents two first-level agents who know each other. If the goals to be achieved by these agents are straightforward, the agents can coordinate their activities just between themselves. The coordination may lie in passing the product that is being assembled from one industrial robot to another in a timely manner. This kind of situation is depicted in Figure 4.3.

Fig. 4.3 Acquaintance model for agents of the first level

Let us suppose that the requirements of a sociotechnical system are represented in the form of a goal model of agent-oriented modelling [47] that was overviewed in Chap. 1 Sect. 1.4. Examples of goal models are depicted in Figures 4.6 and 4.8. If new goals and/or roles are added to the goal tree, more agents may need to be created at the same level and/or at higher levels. In particular, agents of a higher level have to be created if agents of lower level(s) need more coordination to achieve their objectives. In Figure 4.4, an agent of the second level - manager - has been added who interacts with agents of the first level. If a manager is added, agents of the next lower level need to become aware of it because the manager knows a higher-level goal. Another reason for adding a higher level agent may be that in certain situations lower-level agents do not any more manage with the task at hand and need advice by a higher-level agent. For example, if lower-level agents in a sociotechnical system are man-made agents like robots or software agents and the higher-level agent is a human, lower-level agents might ask for his/her advice through a graphical user interface built for this purpose. In this kind of situation all lower-level agents have to be aware of the higher-level agent.

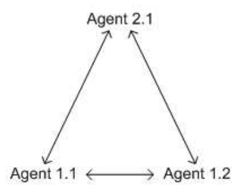


Fig. 4.4 Acquaintance model for agents of the second level

An agent of a higher level might not be aware of (all) the agents at the levels below it. This is illustrated by Figure 4.5, according to which an agent of the third level is aware of just one agent of the second level, who is probably the manager of the second-level agents. Agents of different levels can be added to a sociotechnical system separately and adaptively when system requirements change or goals develop.

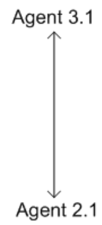


Fig. 4.5 Acquaintance model for agents of the third level

Case study of adaptive and iterative development. We illustrate our considerations by introducing an example from the problem domain of assembling cell phones by a sociotechnical industrial automation system consisting of autonomous robots and humans. The robots are equipped with sensors and actuators and are capable of reasoning and interactions.

We describe the requirements for the sociotechnical system in the form of a goal model. Figure 4.6 shows that for achieving the purpose of the system - Assembling cell phone - several subgoals need to be achieved. Just one agent - an agent playing the Manager role - is aware of the system purpose: Assemble cell phone. For achieving the subgoals, agents playing the six other roles depicted in Figure 4.6 are required. First, an agent playing the Internal Components Assembler role puts together internal components of a cell phone. After that it passes the intermediate product to an Internal Components Tester. If the intermediate product does not pass the tests, an Engineer will fix it and return the product to an Internal Components Tester for an additional iteration of the same tests. After the tests have been passed, an Internal Components Tester sends the intermediate product to a Cover Assembler who equips the cell phone with display and covers. Thereafter it passes the final product to a Cell Phone Tester for ultimate testing. As previously, if a cell phone does not pass the tests, an Engineer will identify and fix the problem. If a cell phone has passed all the tests, it will be forwarded to a Visual Checker for final visual checking. All the roles explained can be performed by either human or man-made agents, such as industrial robots. Please note also that goal models, such as the one represented in Figure 4.6, do not prescribe any temporary sequence of achieving subgoals.

Temporary sequence is present in *design models*, such as the interaction protocol represented in Figure 4.7. Another difference between analysis and design models is that design models represent interactions, behaviours, and knowledge of *agents*

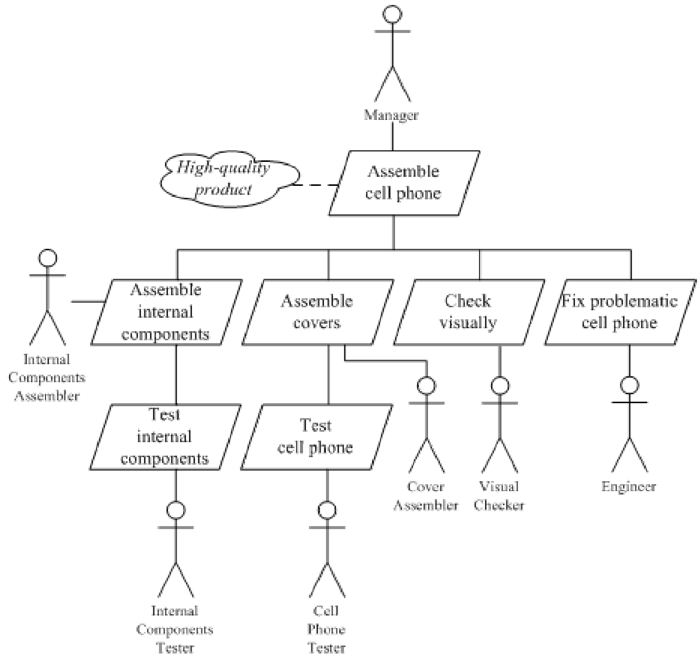


Fig. 4.6 Goal model of assembling cell phones

of specific types playing the roles of the system. The interaction diagram depicted in Figure 4.7 represents that the production process begins with placing internal components of a cell phone onto the printed circuit board. This task is performed by a robot playing the role Internal Components Assembler. After the internal components have been placed onto the circuit board, a human agent playing the role Internal Components Tester tests the internal components. If the internal components pass the tests, the Internal Components Tester will pass the circuit board to the robot performing the role Cover Assembler. If the internal components fail one or more tests, the Internal Components Tester will pass the circuit board to a human agent playing the role Engineer. In Figure 4.7 this case is represented by the first alternative box ”does not pass test”. The box includes the condition which models that the Internal Components Tester keeps sending the circuit board to the Engineer until the circuit board passes the tests. Thereafter the circuit board is forwarded to the robot playing the role Cover Assembler. The robot shields the circuit board and all the other components according to the model specification by the front and back cover and passes the resulting cell phone to the human agent playing the role Cell Phone Tester. The integrity of the cell phone as a whole is tested next. If everything is working properly, the cell phone will be sent to another human agent playing the role Visual Checker who makes sure that the appearance of a cell phone has not been damaged during the assembling process. In case the Cover Assembler has failed to produce a high-quality cell phone and the Cell Phone Tester discovers a problem with it, the newly assembled cell phone will be sent to the Engineer. The Engineer finds and fixes the problem and returns the phone to the Cell Phone Tester. This process continues until the cell phone passes the tests, after which it is again sent to the Visual Checker.

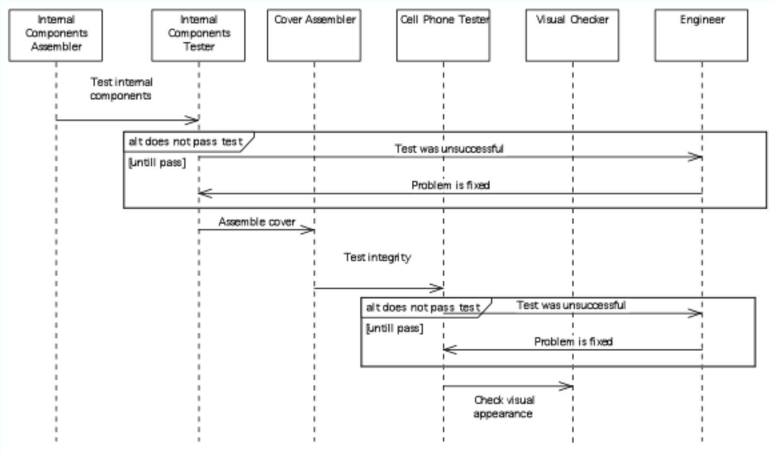


Fig. 4.7 Interaction protocol of assembling cell phones

Let us now suppose that the cell phone industry has new requirements for new cell phones with cameras. A sociotechnical system has to be as flexible as possible to adjust to new processes and incorporate new goals and roles if needed. The production process of cell phones with cameras needs more attention as compared to the production process of ”ordinary” cell phones. In the new requirements described as a goal model in Figure 4.8, this is reflected by the new goal of checking a camera and the new Camera Checker role attached to it. As a result of this change in the requirements, an agent playing the Camera Checker role joins the sociotechnical system. After this, the sociotechnical system continues to function as the production line described in the previous paragraph. The only difference is that an agent playing the Camera Checker role checks cameras before the ultimate visual checking activity is performed. If this new agent can inform the other agents of the system about its capabilities, the interactions, behaviours, and knowledge of just the *affected agents* in the supply chain will change accordingly. There is no need for all the agents of the system to become aware of a new agent playing the Visual Checker role. The whole system functions perfectly well when the new agent knows what to do and who to interact with and only some agents are aware of the new agent. In a similar way, all agents do not have to be aware of the overall goal of the system. For example, a Visual Checker does not have to know that the purpose of the system is to assemble cell phones. The system operates very well when a Visual Checker only knows how to control cameras and who to interact with.

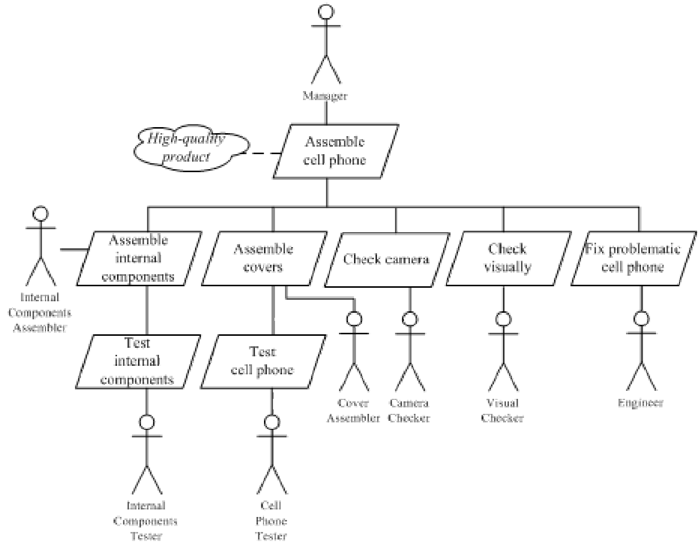


Fig. 4.8 Goal model of assembling cell phones with cameras

Figure 4.9 depicts the interaction model of assembling cell phones with cameras. In comparison with the interaction diagram represented in Figure 4.7, it includes another alternative box entitled ”camera does not fit”. The box models that if the camera is improperly placed or contains dust, a human agent playing the role Camera Checker will send it to the Engineer. This alternative process is repeated until the Engineer has fixed all the problems with the camera that have been detected.

Bottom-up iterative design organized in the way described in this section results in *evolutionary* sociotechnical systems. In such systems, all the agents do not have to know what each of them knows. A sociotechnical system functions properly when each agent is aware of its own objectives as a minimum and about the means of achieving them.

In contemporary complex sociotechnical systems it is not feasible to possess all the information about the environment and to keep this information continuously updated. To reflect this, we proposed in Sects. 4.5 and 4.6 of this Chapter an iterative bottom-up development approach of sociotechnical systems. Such iterative development is flexible and adaptive. Therefore it is easy to adapt to a rapidly changing environment. In the near future, we plan to complement goal models by role and domain models, and interaction models by agent behaviour and knowledge models and to design and implement an environment that would enable to try our approach out in series of experiments. Designing the environment comprises working out a formal language for

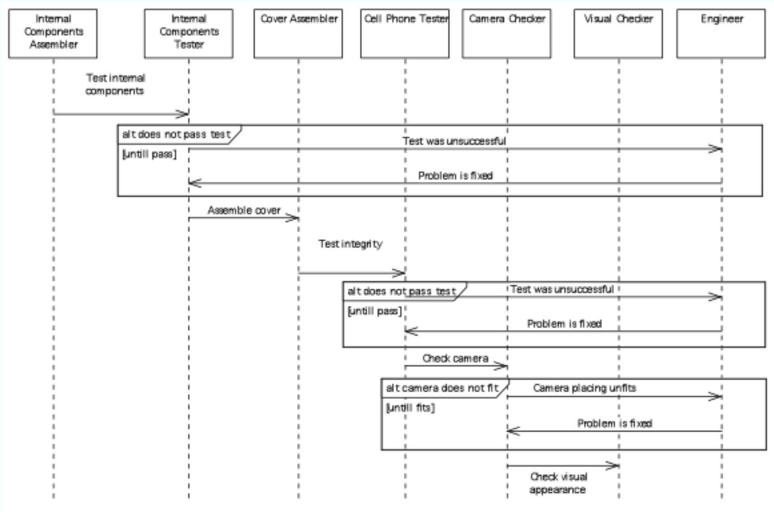


Fig. 4.9 Interaction protocol of assembling cell phones with cameras

4.7 A role evolution mechanisms as an information source of trust

In Open Multi-Agent Systems (OMAS), deciding with whom to interact is a particularly difficult task for an agent, as repeated interactions with the same agents are scarce, and reputation mechanisms become increasingly unreliable. Here we present a mechanism which can be used by agents in an OMAS to take more informed decisions regarding partner selection, and thus to improve their individual utilities. This mechanism monitors the interactions in the OMAS, evolves a role taxonomy, and assigns agents to roles based on their observed performance in different types of interactions. This information can be used by agents to better estimate the expected behaviour of potential counterparts in future interactions. We thus highlight the descriptive features of roles, providing expectations of the behaviour of agents in certain types of interactions, rather than their normative facets.

In decision making (DM) processes for selecting partners agents may make their choice supported on three different types of information, namely: *i)* past own experience; *ii)* opinions from neighbours (reputation); and *iii)* other "organisational" information sources. The first two types have already been widely studied in many works [31, 48]. Some other works [27] have studied how organisational information influences agents' selections, especially when no direct experiences – or not reliable enough – have been collected before.

We deal here with this third type of information aforementioned, namely how agents can use organisational structures to better determine "good" partners to interact with, especially if no valuable direct experiences are available to reason about. We show that agents cannot only exploit existing organisational structures, in particular, role taxonomies, to determine trustworthy candidates to interact with, but we also put forward a mechanism that makes use of the information managed by the agents' trust models so as to create and evolve role taxonomies. We claim that this taxonomy evolution provides agents with more precise information, helping them to make better decisions such as to decide which other agents to interact with. Thus, in [11] it is proposed an adaptive mechanism that evolves role taxonomies by using a multidimensional clustering algorithm to capture behavioural patterns among agents.

Organisational Structures for Agents DM. The environment we use to describe the mechanism presented in this work is based on Task-oriented Multi-Agent Systems (T-MAS) which can be specified as follows:

Definition 2 *A T-MAS is a tuple TM=*$\langle \mathscr{A}_g, \mathscr{X}, \mathscr{T}, \mathscr{U} \rangle$*, where:*

- $\mathscr{A}_g$ *is a set of agents participating in the MAS; we assume each agent* $a \in \mathscr{A}_g$ *has an utility function* $\mathscr{U}_a : \mathscr{A}g \times \mathscr{T} \rightarrow R$*, where* $\mathscr{A}g$ *is the delegated agent that performs the task* $\mathscr{T}$*;*
- $\mathscr{X}$ *is the environmental state space;*
- $\mathscr{T}$ *is a set of tasks that can be performed by agents;*
- $\mathscr{U} : \mathscr{X} \rightarrow R$ *is the system utility function;*

The functioning of a T-MAS is as follows (at each time step): i) a task is assigned to each agent $a_1 \in \mathscr{A}_g$; ii) if an agent a_1 cannot perform the task by itself it reassigns (delegates) the tasks to another agent $a_2 \in \mathscr{A}_g$; and iii) agents a_2 performs the task and a_1 obtains a utility from the performance. Furthermore, we assume that the utility obtained by an agent at a certain time step is equivalent to the agent's perception on the fulfilment of the delegated task to another agent. Note that this definition of individual utility allows for *subjective* utility functions. In this sense, $\mathscr{U}_a(b,t)$ represents the subjective perception of agent a on how well agent b performs task t. Notice that an agent may delegate a task to itself if considers that it is the more qualified agent to carry it out.

Organisational Information. The mechanism presented in [11] is based on the use of the concepts *role* and *role specialisation taxonomy*. We conceive roles from the point of view of an observer, i.e. as a set of *expectations* regarding the behaviour of agents performing certain actions. This means that a role generates by itself some public expectations over certain actions that agents playing it should accomplish.

A *role* in a T-MAS is a pair $\langle r, \mathscr{E} \rangle$ so that the agents playing the role r are qualified to perform the tasks in the set $\mathscr{E}$ in the sense that they are "skillful" for those tasks.

A role specialisation taxonomy structures the roles by establishing a specialisation relation $\triangleright_r$ based on the skills of the agents playing those roles; that is, given two different roles $r_1, r_2 \in R$ then $r_1 \triangleright_r r_2$ iff. there is a subset of tasks from r_2 on which agents playing role r_1 perform better, on average, than agents playing role r_2. The hierarchy contains a top role - the root of the taxonomy $\langle r_{root}, \mathscr{E}_{root} \rangle$ - which contains all tasks and is not a specialisation of any other role. This is consistent with the assumption that every agent can perform every task. We can assume that every agent in a T-MAS plays at least the top role.

A Trust Model for Agent's DM. A trust model is usually used to endow agents with an internal representation of information about others in order to better choose partners to interact with in any DM process. In the context of a T-MAS, we use the notion of trust model as a mechanism that drives the agent to choose the most trustworthy agent to which it can delegate a given task. Trust models aims at calculating expectations on other agents on particular situations, by either using past information gathered through the time – based on past interactions – or inferring using opinions from third party using their own previous assessments.

The main contribution of the work is twofold: i) building role taxonomies containing on the expectations that the agents participating in the T-MAS are currently calculating during their execution in the T-MAS; and, ii) agents may make use of the created role taxonomies in order to tune up their own expectations on different situations. These two processes are executed in parallel and continuously repeat during the T-MAS lifetime.

Next algorithm describes how an agent a uses the information provided by a role specialisation taxonomy $\mathscr{RT}$ together with its own experience about previously delegated tasks in order to select an appropriate agent to which it can delegate a given task t.

1. $r = mostSpecializedRolesForTask(t)$
2. $\mathscr{A}_x = agentsPlayingRoles(r)$
3. $bestAgent = localTrustEvaluation(\mathscr{A}_x, r, t)$
4. $delegate(t, bestAgent)$

For the calculation of trust values $t_{a \rightarrow \langle a_i, r_k \rangle} \in [0..1]$, we assume that agents store their past experiences in their internal structure in form of confidence values $c_{a \rightarrow \langle a_i, r_k \rangle}$, denoting the recompiled confidence an agent a has in agent a_i playing role r_k .

Evolution of Role Taxonomies. Creation of new roles is be based on trust that other agents have on a specific role - that is similar to say "on the agents playing that role in the system". Trust is a subjective measure, since not all agents neither have to share the same preferences in the system nor have to use the same trust model. The mechanism defined in [11] tries to build a source of information - role taxonomy - from subjective individual assessments of trust.

This mechanism employs clustering methods to capture behavioural patterns of agents performing tasks. The idea is to identify groups of agents that perform a set of tasks better than others and to reflect such cases in form of a new role. In order to do this it is assumed that agents store confidence values $c_{a \rightarrow \langle a_i, t \rangle}$, representing agent a's recompiled experience on how well agent a_i performs a task t (from its particular point of view). The confidence values stored by agents provide a means to represent agents as a point in the n-dimensional vector space formed by all possible tasks $t \in \mathscr{T}$ in the T-MAS where n is the number of tasks in $\mathscr{T}$. In particular, each agent a_i can be represented as a tuple $\widehat{a} = (c_1, c_2, \ldots, c_n)$ where c_k is defined as follows:

$$c_k = \frac{\sum_{a \in \mathscr{A}_g} c_{a \rightarrow \langle a_i, t_k \rangle}}{|\mathscr{A}_g|} \tag{4.2}$$

The set of vector representations of agents – e.g., the trust space formed by agents – is denoted by $TS = \{\widehat{a} = (c_1, c_2, ..., c_n) | a \in \mathscr{A}_g\}$. In a similar way, given a role $r_k \in \mathscr{R}$, a trust space for the agents that have ever played that role is defined as: $TS_{r_k} = \{\widehat{a} = (c_1, c_2, ..., c_n) | a \in \mathscr{A}_g \, and \, a \, enacts \, r_k\}$.

Trust-based Multidimensional K-Means. To specialize roles - create new roles in the role taxonomy - the *K-means* clustering algorithm can be applied, where k represents the number of clusters to be created in each execution. Let TM be a T-MAS with a set of roles R and a role specialisation taxonomy $\mathscr{RT} = (R, \triangleright_r)$. In order to evolve the role taxonomy, the clustering algorithm is applied to each set TS_{r_j} with $r_j \in R$ and r_j being a leaf in the taxonomy $\mathscr{RT}$. On each execution, the algorithm returns a set of k clusters. A cluster centroid represents the expected behaviour of all the agents belonging to it and the whole cluster represents a pattern of behaviour for all the agents included.

The possible clusters returned by the algorithm are candidates for the creation of new roles. We process the clusters and only convert it into a new role r_x if the agents enacting r_x provide a better performance (on average) on at least one of the tasks of the role it extends. Furthermore, when deciding whether a cluster should form a new role or not, the mechanism applies two additional criteria: (a) we do not create roles with "bad" behaviours. We apply a threshold θ such that a new role is only created if the tasks it specializes have at least an expected value of θ; (b) in most of the cases we would want to create new roles if, in fact, they may have a "long" life. That is, most of the times there is no much sense on creating roles when only an agent may play it. Would make sense to create role *Surgeon* if only one agent in the world could play it? For that reason, we include another threshold, called Υ that determines the minimum number of agents that a cluster must include to have the possibility of converting the cluster into a new role.

4.8 Group-Oriented Coordination

Adaptive Agent Organisation can focus on different perspectives: the macro-level for analysing and coordinating the overall performance of an IT-ecosystem, and the micro-view for observing and manipulating the autonomous agents and an interaction layer for interlinking both. Taking global and individual objectives into account, the metaphor of groups can combine them to improve their utilities and benefits. Co-operation and coordination mechanisms need to find an equilibrium for global and individual objectives. We apply the group-oriented coordination on a simple example allowing agents to form faster and slower groups, described more in detail in Görmer&Müller [25]. As previously explained in Chap. 1 Sect. 1.5, *Grouping* allows an agent to extend its range of perception (RoP) by exchanging information with other members. Agents are coordinated at group level. Group-oriented coordination allows agents e.g. to form faster and slower agent groups. Each group has an agent group leader. In case fast groups are blocked by other slow groups, the group leaders will communicate with each other to arrange plans for each group (called group plans). The group plans are known to all group members. Acting based on group plans, quick agents can avoid being blocked by other slow agents and vice versa. Informally, the three main elements of group-oriented coordination can be described as follows:

1. **Decentralised dynamic agent grouping**: Agents autonomously form groups desiring the same goal. An agent group contains a group leader and members. The group leader is responsible for the coordination of the group's members to avoid detected *conflict situations* with other groups or agents. Since agent organisations usually are situated in dynamic environments, agent groups are dynamically created and maintained. This means, the number of agent groups and the number of members of a group change constantly over time.
2. **Conflict detection and global coordination**: The second element of group-oriented cooperation is to coordinate members in case of conflict between agent groups. A conflict situation can be detected by group leader or members. Each agent of a group will scan in its range of perception (RoP) for other groups, which potentially will block its group (conflict group detection). Once a conflict situation between two or more groups is detected, it will be communicated to group leaders. A group leader coordinates its members by defining an appropriate group plan. The choice of group plans is a negotiating process between leaders of groups, which are in conflict situation. The group plans have a warranty that members are not blocked by members of other groups. Communication limitations only allow an agent to communicate with other agents when they are in a fixed RoP. This means, a leader cannot exchange message with another leader of a conflict group if the two are out of communication range. However, we assume that members of an agent group can forward messages of their leaders to receivers in a multihop fashion.
3. **Coordination strategy of an individual agent**: At this step agents decide their plan of actions for the next time period. Reaching and maintaining desired goal is the original goal of each agent. An agent chooses its plans, which allows it to reach its goal as soon as possible. However a member agent should (sometimes) obey the coordination of its group leader to avoid conflict situations. Thus, an agent should always decide whether to choose its own plan based on the coordination of leader or to choose it based on its local goal.

4.9 Organisational perspective of a task assignment model for cooperative MAS

The problem of coordination of multiple mobile agents which collaborate to achieve a common goal in an environment with variable communication constraints arises in numerous man-made systems. In order to analyze such systems, the design of coordination and agreement strategies and mechanisms with specific inter-agent information exchange principles, and limited communication, together with their influence to the emergent behaviour of the system must be addressed.

In [24] we address a cooperative control problem in which a team of mobile agents under different inter-agent communication conditions has to accomplish certain mission. Generating the individual agent trajectories and associated actions that accomplish this objective can be viewed as the dynamic assignment of each agent to certain subset of spatially distributed tasks in some chronological order.

To efficiently assign agents to tasks, we are interested in finding a maximum matching (i.e., a one-to-one assignment of agents to tasks) which minimizes some multi-agent system's (MAS) collective cost function. The value of the latter is assumed to be the sum of the individual costs associated with each agent-task pair matching in each assignment run, depending on some factor (e.g., time, energy, etc.) that it takes every agent to travel to and complete its assigned task. This kind of problem is equivalent to the minimum weight maximum matching problem in a bipartite graph or assignment problem in the operational research field; the latter can be written as an integer linear program and optimal solutions can be computed in polynomial time. Many centralized algorithms of polynomial complexity exist to solve it, e.g., primal simplex methods, Hungarian, dual simplex (see, e.g., [37]) and relaxation methods (see, e.g., [30]).

However, in the case of decentralized cooperative MAS where there is no centralized decision-maker and each agent keeps potentially different local information, the centralized algorithms for task assignment are inadequate. Since, generally, agents are placed on different positions and possibly with different utility functions, the benefit and the costs of getting assigned to a particular task will be different. Assuming that agents are capable of communication, and that each agent may have information that is local and not known globally throughout the team, agents will have to exchange relevant information and negotiate in order to find the sufficiently good assignment for all. Such MAS scenarios require decentralized coordination mechanisms and rules which will assign tasks to appropriate agents in order to obtain a mutually acceptable and efficient outcome. In [24], a distributed coordination model for task assignment is proposed, which is based on two coordination mechanisms which are complementing one another based on the shape of the communication graph among agents: a distributed version of the Hungarian method [24] which calculates an optimal solution to the task assignment problem, and the dynamic iterative auction [35] inspired by Bertseka's auction algorithm. Agents select the task assignment mechanism based on the connectivity of their communication graph, and when selected, the mechanism defines a set of roles and the strategies for these roles that by mutual interaction find the multiple task assignment that maximizes the global system's utility. It is clear that the shape of the communication graph is directly influenced by the choice of the agents' transmitting range, i.e., the larger the range, the less likely it is that the communication network becomes disconnected.

The proposed task-assignment model integrates two mechanisms for efficient task assignment: distributed algorithm based on the Hungarian method [24] in the case of complete communication graph and the distributed iterative auction algorithm [35] inspired by Bertsekas auction algorithm [10] in the case of a disconnected communication graph among agents. They are integrated in this way because the former is less computationally expensive and together with the latter, it gives the optimal assignment solution in the case of completely connected communication network. The latter can function also in the case of disconnected communication network, resulting in the sub-optimal result where the performance is bounded in the worst case ([24],[35]).

These two coordination mechanisms promote desirable social behaviour in terms of efficient optimal or close to optimal task assignment solutions for collaborative organisation-based multi-agent systems (MAS) with variable inter-agent communication range. These two mechanisms complement each other depending on the momentary communication range among agents.

The result is a joint plan that is optimal, or sub-optimal regarding the global utility of a MAS. The mechanisms are stable, informational decentralized and efficient in respect to the information exchange in the sense that agents communicate small amounts of relevant information in each round of the performance instead of completely specifying their preferences over the entire space of future actions and possible events. The distributed Hungarian method is much less computationally expensive than the auction algorithm which, in contrary can be used also when the communication graph among agents is not fully connected. The lack of information in unconnected communication graph results in an inferior but still acceptable assignment result. We applied the model in the organisation-based ambulance management of patient emergencies. In this scenario, ambulances act as a team that has the objective to reach each appearing emergency patient in the shortest time possible. Obviously, patients might appear in different times and places. Then, the task of the ambulance team is to organize its operation by reaching an agreement on ambulance-patient assignment. In the scope of our model, patients are seen as tasks. We assume a decentralized scenario since ambulances are intrinsically decentralized resources, i.e., each ambulance crew can control only its local behaviour and can only exchange information by communication with other agents in the emergency management system. It is assumed that each ambulance agent has an information regarding its position and can receive the information regarding the position of all patients in the environment through the coordinates on a map of the environment. If the number of patients is small, a patient assignment problem can be solved by a centralized emergency manager. If the latter is missing, then the ambulance agents, by mutual communication and information exchange, find an optimal assignment solution through the distributed Hungarian method algorithm. When the connectivity of the communication graph is not complete, the agents can follow the dynamic iterative auction algorithm with mobility to get assigned and manage the emergency patient cases in a decentralized manner. More details of the application of distributed task-assignment model presented here can be found in the Part on Applications of Agreement Technologies in this book.

4.10 Conclusions

Organisations represent an effective mechanism for activity coordination, not only for humans but also for agents. Nowadays, the organisation concept has become a relevant issue in the multi-agent system area, as it enables analysing and designing coordination and collaboration mechanisms in an easier way, especially for open systems.

In this chapter, we have presented different approaches for adaptive agent organisations, including methods for designing and/or implementing this kind of systems. In all these sections we have emphasized the proposals developed within the COST action IC0801.

References

1. Aldewereld, H., Penserini, L., Dignum, F., Dignum, V.: Regulating Organizations: The ALIVE Approach. In: Proceedings of ReMoD (2008)
2. Aldrich, H.: Organizations evolving. Sage Publications Ltd (1999)
3. Aldrich, H.: Organizations and environments. Stanford Business Books (2007)
4. Allert, H., Richter, C.: Practices, systems, and context working as core concepts in modeling socio-technical systems. In: Proceedings of the Fifth International Workshop on Philosophy and Informatics, WSPI 2008. Kaiserslautern, Germany (2008)
5. Andersson, J., de Lemos, R., Malek, S., Weyns, D.: Modeling dimensions of self-adaptive software systems. In: Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 27–47. Springer (2009)
6. Argente, E., Botti, V.J., Carrascosa, C., Giret, A., Julián, V., Rebollo, M.: An abstract architecture for virtual organizations: The THOMAS approach. Knowl. Inf. Syst. 29(2), 379–403 (2011)
7. Asimov, I.: I, Robot. Gnome Press, New York (1950)
8. Barnett, W., Carroll, G.: Modeling Internal Organizational Change. Annual review of sociology 21 (1995)
9. Bernon, C., Gleizes, M.P., Peyruqueou, S., Picard, G.: ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering. In: ESAW'02 (2002)
10. Bertsekas, D.: The auction algorithm: A distributed relaxation method for the assignment problem. Annals of Operations Research 14, 105–123 (1988)
11. Centeno, R., Hermoso, R., da Silva, V.T.: Role evolution in open multi-agent systems as an information source for trust. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010). ACM Press, Toronto, Canada (2010)
12. Cuesta, C.E., Perez-Sotelo, J., Ossowski, S.: Self-organising adaptive structures: the shifter experience. ERCIM News 85, 35–36 (2011)
13. DeLoach, S., Oyenan, W., Matson, E.: A capabilities-based theory of artificial organizations. J. Autonomous Agents and Multiagent Systems 16,13–56 (2008)
14. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. PhD thesis, Utrecht University (2003)
15. Dignum, V., Dignum, F.: Towards formal semantics for reorganization. Technical report UU-CS (2006)
16. Dignum, V.: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models. Information Science Reference (2009)

17. Dignum, V., Dignum, F.: A logic for agent organizations. In: Proceedings of the 3rd Workshop on Formal Approaches to Multi-Agent Systems (FAMAS) (2007)
18. Dignum, V., Dignum, F., Furtado, V., Melo, A., Sonenberg, L.: Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agents Societies. In: Proc. Socially Inspired Computing, AISB Convention, volume 230 (2005)
19. Esparcia, S., Argente, E.: Forces that drive organizational change in an Adaptive Virtual Organization. In: Proc. CISIS-2012 Conference. p. In Press (2012)
20. Galloway, A.: Protocol: How Control Exists after Decentralization. The MIT Press (2004)
21. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994)
22. Gardelli, L., Viroli, M., Casadei, M., Omicini, A.: Designing self-organising environments with agents and artefacts: a simulation-driven approach. International Journal of Agent- Oriented Software Engineering 2(2), pp. 171–195 (2008)
23. Gazendam, H., Simons, J.: An analysis of the concept of equilibrium in organization theory. In: Proceedings of the Computational and Mathematical Organization Theory Workshop (1998)
24. Giordani, S., Lujak, M., Martinelli, F.: A distributed algorithm for the multi-robot task allocation problem. In: Trends in Applied Intelligent Systems: Proc. of the 23rd internat. conf. on Ind. eng. and other applications of applied int. systems. pp. 721–730. Lecture Notes in Artificial Intelligence, Springer-Verlag (2010)
25. Görmer, J., Müller, J.P.: Multiagent system architecture and method for group-oriented traffic coordination. In: Proceedings of IEEE DEST 2012. IEEE press
26. Haesevoets, R., Weyns, D., Holvoet, T., Joosen, W.: A formal model for self-adaptive and self-healing organizations (2009)
27. Hermoso, R., Billhardt, H., Centeno, R., Ossowski, S.: Effective use of organisational abstractions for confidence models. In: ESAW. LNAI, vol. 4457, pp. 368–383. Springer (2007)
28. Hoogendoorn, M., Jonker, C.M., Schut, M.C., Treur, J.: Modeling centralized organization of organizational change. Computational and Mathematical Organization Theory, 13(2), pp. 147–184 (2007)
29. Hubner, J., Sichman, J., Boissier, O.: Using the Moise+ for a Cooperative Framework of MAS Reorganisation. Advances in Artificial Intelligence–SBIA 2004, pp. 481–517, Springer (2004)
30. Hung, M., Rom, W.: Solving the assignment problem by relaxation. Operations Research 28, 969–982 (1980)
31. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: Fire: An integrated trust and reputation model for open multi-agent systems. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI) (2004)
32. Kota, R., Gibbins, N., Jennings, N.R.: Self-organising agent organisations. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pp. 797–804. International Foundation for Autonomous Agents and Multiagent Systems (2009)
33. Kramer, J., Magee, J.: Self-managed systems: an architectural challenge. In: Future of Software Engineering (FOSE@ ICSE2007). pp. 259–268. IEEE (2007)
34. Luhmann, N.: Social Systems. Stanford University Press, California (1995)
35. Lujak, M., Giordani, S.: On the communication range in auction-based multi-agent target assignment. In: IWSOS11: Proc. of the 5th internat. conf. on Self-organizing systems. Lecture Notes in Computer Science, vol. 6557, pp. 32–43. Springer-Verlag (2011)
36. McShane, S., Travaglione, T., Bazley, M., Hancock, P., Hill, C., Berk, J., DeMarzo, P., Stone, R., Tichy, N., Bennis, W., et al.: Organisational Behaviour on the Pacific Rim+ E Book. Prentice Hall International Series in Industrial and Systems Engineering The Guild Village, Hackett Entry 2, Hackett Drive, Nedlands WA 6009
37. Nash, S., Sofer, A.: Linear and nonlinear programming, vol. 692. McGraw-Hill, New York (1996)
38. OSGi: formerly known as the open services gateway initiative, now an obsolete name., `http://www.osgi.org/`

39. OVAMAH: Ovamah - organizaciones virtuales adaptativas: Técnicas y mecanismos de descripción y adaptación, `http://gtia-ia.dsic.upv.es:8080/ovamah`
40. Perez-Sotelo, J.S., Cuesta, C.E., Ossowski, S.: The agreement as an adaptive architecture for open multi-agent systems. In: II Taller de Sistemas Autonomos y Adaptativos (WASELF 2009). vol. 3, pp. 62–76. SISTEDES, San Sebastian, Spain (2009)
41. Picard, G., Hubner, J.F., Boissier, O., Gleizes, M.P.: Reorganisation and Self-organisation in Multi-Agent Systems. In: 1st International Workshop on Organizational Modeling, p 66 (2009)
42. Popper, K.: The Poverty of Historicism. Harper and Row, New York (1964)
43. Prigogine, I.: The End of Certainty. Time, Chaos and the New Laws of Nature. The Free Press, New York (1997)
44. Prokopenko, M., Boschetti, F., Ryan, A.: An information-theoretic primer on complexity, self-organization, and emergence. Complexity 15, 11–28 (2008)
45. Ramirez, A.J., Cheng, B.: Design patterns for developing dynamically adaptive systems. In: ICSE2010-SEAMS. pp. 49–58 (2010)
46. Robertson, R.: Globalization: Social theory and global culture, vol. 16. Sage Publications Ltd (1992)
47. Sterling, L., Taveter, K.: The Art of Agent-Oriented Modeling. The MIT Press, Cambridge, MA, and London, England (2009)
48. Teacy, W.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)