Application Domains

Julian Padget[1], Huib Aldewereld[2], Pablo Noriega[3], and Wamberto Vasconcelos[4]

[1] University of Bath, United Kingdom
jap@cs.bath.ac.uk
[2] Delft University of Technology, The Netherlands
h.m.aldewereld@tudelft.nl
[3] Intitut d'Investigació en Intel·ligència Artificial, Barcelona, Spain
pablo@iiia.csic.es
[4] University of Aberdeen, United Kingdom
w.w.vasconcelos@abdn.ac.uk

Abstract. The subject of this chapter is the application of models for social coordination (M4SC). Deriving from and extending multiagent systems (MAS), applications of M4SC inherit the attributes of MAS, such as multiple actors, distribution, and heterogeneity, but bring characteristics of their own that are essential to social coordination over and above conventional MAS. In this chapter, we first discuss these differentiating properties of M4SC, identifying essential and non-essential (but commonly occurring) characteristics. This we follow with a wide-ranging (but not exhaustive or exclusive) discussion of classes of M4SC applications, which are supported by structured, illustrative summaries of noteworthy examples of applications, created using the frameworks described elsewhere in this volume. Finally, we give an overview of a broader range applications developed with the frameworks, with links to further reading.

1 Introduction

The ostensible purpose of models for social coordination (M4SC) is to provide the support for an effective coordination of autonomous entities, thus, in a very crude sense one may argue that M4SC are designed to support a class of multiagent systems. However, one may identify distinguishing features of the class of multiagent systems that one can build with M4SC.

As we will see in the examples below, the variety of systems that have been built using M4SC is large but we may find in them some features that are supported by all the frameworks discussed in this book. We claim these features are the ones that discriminate applications of M4SC from the more generic MAS applications. Other features are not essential in this sense, although —as the examples illustrate— they are rather frequent in the systems that are built with the frameworks presented in this volume. A quick discussion of those essential and contingent features should allow for a richer appraisal of the examples we describe in this chapter.

2 Characteristics

Essential characteristics are, we contend, exhibited in all applications of models of social coordination, and comprise:

- **Institution-driven design:** Models for social coordination are inspired by the classical notion of *institution*, advanced by [70,85], as a way of establishing the ground rules for articulating social coordination. The objective is to provide support for the specification, enactment and the evolution of the system over time, so that resulting systems can be resilient and robust to (un)expected deviations, without external intervention. Applications of M4SC are a distinct form of MAS, in which the three following fundamental institutional aspects may be found: (i) The establishment of the ontological constraints (what are the entities, meanings, effects) upon which the system is defined and works; (ii) the provision of governance devices that support the control or coordination mechanisms that are imposed on interaction; (iii) the availability of a dynamic *institutional state* – involving the facts that hold, the events that happen, and the changes these produce – that is common to all participants.
- **Independent aims of actors:** Applications of models for social coordination differ from traditional MAS in that they are by definition heterogeneous. While MAS work well with either homogeneous or heterogeneous actors, models for social coordination are most advantageous in heterogeneous settings. Where heterogeneity presents problems for MAS (*e.g.*, coordination, expectation, predictability), models for social coordination are intended to provide the answer to these problems. A particular aspect of heterogeneity in the application domains of social coordination is the fact that there are multiple actors (one might even call them stakeholders) and there is an continuous struggle between the interests of the actors individually and those of the society as a whole. This means that each of the actors has to solve a tension between: (i) their own needs (*self-motivation*), (ii) the needs of the community/society (*collective goals*), and (iii) coordination and control imposed on them (*control*).
- **Opacity of actors:** Models of social coordination address distributed solutions where the sophistication of individual actors' design and the very large number of actors make it impossible to expose and process their internals (*i.e.*, the information and knowledge individual actors store, their perception of the environment, and their "inner workings"). Models for social coordination must offer means to expose features of actors selectively, thus helping designers concentrate on essential aspects of coordination. Adopting the software agent view [94] where there is a clear separation between actors and their (virtual) environment, the basic assumption is that designers (should) only have access to a record of the (effects of the) actions of actors. Typical (effects of) actions are, for instance, messages exchanged, positions of individuals in a team of robots, or updated records in a database. Some models differentiate between internal and external actors: the former are part of the architecture or infra-structure (providing operational support) and the latter are third-party entities, heterogeneous in nature and conferring openness to the overall distributed solution. Although it is technically possible to make internal agents fully transparent, it is doubtful if models and methodologies would be useful, given the numbers and sophistication of actors.
- **Explicit (representation of) regulation:** Coordination always requires some form of control and the approach predominant in models for social coordination is through explicit representation of regulation that is *external* to the minds of agents. The crit-

ical contributions delivered by explicit regulation are: (i) the system can function in ways not perceived at design time, yet still be constrained as a whole, while the implementation of regulation allows varying degrees of flexibility around different aspects of behaviour (ii) (system) transparency is intrinsic (iii) system-level outcomes are predictable, while, necessarily, agent outcomes are not, which is in direct contrast to the traditional notion of openness in MAS, where it is knowledge of (the inner workings of) each agent that leads to predictability/guarantees of the system('s performance). The implementation of regulation (*e.g.*, enforcement, constraint, sanction etc.) varies from system to system, reflecting a spectrum of intentions from 'could', through 'ought' to 'must' or less informally from option, through recommendation to protocol. In general, agents can work with multiple (differing) forms of regulation and mechanisms (to reason with and about the regulations in their decision making). Explicit representation of regulation can introduce a different level of cooperation in M4SC applications than that typically seen in MAS applications. Cooperation here is not in the agent-oriented sense of benevolent vs. self-interested, but rather cooperation that is induced through regulation in order to achieve institutional goals.

Nonessential characteristics are typical but not necessary characteristics of applications of models for social coordination:

- **Complex:** A complex system is typically characterised by (system) behaviour that is not predictable from models of its constituent parts and which typically arises from the interaction between those components. Analytical solutions are often used to explore the properties of complex systems but scaling (e.g. moving from 2 actors to n) can mean an analytic solution no longer exists or lead to a loss of fidelity through modelling at a higher degree of abstraction. Empirical solutions scale readily and permit arbitrary degrees of fidelity, but modelling mistakes are not easily detected and verification and validation requires great care. The use of regulation (q.v.) as a means to capture, formalize and operationalize requirements can be especially helpful in building the bridge between intentions and outcomes to increase confidence in empirical solutions.
- **Hybrid, technology-agnostic, human/software systems:** Models of social coordination address a growing class of sophisticated distributed systems comprising hundreds, perhaps thousands, of components. These components can be software or humans, giving rise to *hybrid socio-technical systems*. Such systems, due to their size and sophistication, will likely integrate software components developed by many parties, implemented in different programming languages, and adopting disparate technologies and frameworks. These systems need to be studied in a way in which such differences – human vs. software, reactive vs. proactive actors/agents, disparate technologies, and so on – can safely be abstracted, thus helping designers and engineers to handle their complexity better. A significant concern is to support *interaction* among components/actors, ensuring individual (actor-specific) and global (system-wide) goals can be achieved. Governance of socio-technical systems poses important ethical issues, including, for instance, if a distinction should be made among human and software actors (and if so, how so), and what provisions should be in place to define expectations and trust among the actors.

– **Online/real-time:** Applications of M4SC are designed to inter-operate with the physical world, through the perception of exogenous events, perhaps raw, but more likely cleaned, processed and possibly aggregated and through action, perhaps via actuators, but more likely the presentation of information in some form to human participants. The multiple levels through which data is passed (in both directions) via processes that cannot necessarily offer hard response time bounds, means such a system is not "real-time" in the sense it is conventionally understood, but should nevertheless be capable of *timely* interaction within the context of the social institutions that govern the system as a whole.
– **Dynamics of agent population:** One of the objectives of **regulation** is to make the system accessible to all sorts of agents, not just those imagined at design time. A corollary of this property is that the system may not make assumptions about which roles are (in)active or which agents are present/absent at any given time, thus allowing it to respond correctly to the dynamics of the agent population.
– **Revision:** Changing environment and/or changing participant behaviour can result in an institutional framework being no longer appropriate, in that it does not meet the needs of its participants. Revision might be brought about by institutional means such as proposals for rule change and voting, but would be beyond the reasoning capacity of most current agent architectures. Alternatively, an external agency observing patterns of behaviour might impose institutional change based on a notional performance metric. Both of these scenarios pertain to live systems and raise the challenging issue of how to handle the transition between old and new rules. Finally, there is the case of institutional incompatibility, which can be manifested as permission/prohibition or obligation/prohibition conflicts. These may not be apparent from inspection, but can be identified automatically and either resolved manually by designers or (semi-)automatically using techniques such as inductive logic programming.

3 Exemplars

The characteristics set out in the previous section aim to show that M4SC are concerned with the design and implementation of the environment and not with the internals of participating agents. This perspective fosters: (i) a sharper focus on coordination and governance features; (ii) a separation of concerns between the design of social conventions and the design of the capabilities of individuals; and (iii) an encapsulation of best practices and procedural conventions through formal models expressed as one or several, cooperating or interacting instances of institutions. Some M4SC allow for the coexistence of several active institution instances at any time and, more significantly, the possibility for any agent to be active, simultaneously, in more than one.

Interactive systems, where wrong-doing of individual agents may have significant cost, may require a trustworthy third party and strong conventions. M4SC is aimed at the kinds of systems that require a more flexible form of control, for which purpose an explicit presentation of normative statements as well as corresponding normative features for detecting misbehaviour and enforcing norms (in a variety of ways) are the defining characteristics. To illustrate this point further, consider the classes of:

- Regulated systems whose conventions may be fixed but where the consequences of those conventions are difficult to evaluate without some systematic experimentation. For example, mechanism design [7,79], and public policy management [39].
- Regulated systems whose debugging or actual use involves a mixed population of human and software agents. For example, on-line multiplayer games [59], participatory simulation, and on-line dispute resolution [79].
- Coordination systems where, in spite of the need for a sharp specification of interactions, the precise flow of activity may be difficult to perceive in advance, or multiple variants of some activities may be necessary at some point during the system life-span. For instance, supply networks [4,95], social opinion gathering [82], and web-service choreographing [5].

Finally, to provide concrete in-depth examples of how the essential and non-essential characteristics can occur in practice, we identify a variety of categories of application domains that exhibit requirements suited to M4SC characteristics:

- **Markets**: convention-driven e-business environments, like auctions.
- **Mirror worlds/serious games/augmented reality**: virtual environments where humans and agents interact.
- **Social simulation**: systems to test, investigate, or evaluate social theories.
- **Policy making**: systems to analyse the evaluation, revision, prosecution, monitoring, and lifecycle of policies.
- **Flexible workflows**: systems to dynamically adapt processes according to the situation, including service composition, workflow governance, flexible business processes.
- **Critical infrastructure analysis**: systems to evaluate, monitor, or test the (flexibility) of critical infrastructures, to assist where testing in real-life situations might prove difficult or unfeasible.
- **Collective decision-making**: systems to assist users or user groups to reach a consensus about (shared) issues and topics.

The relationship between these categories and the (non-)essential characteristics set out in the previous section is shown in table 1. Moreover, all the domains, it can be assumed, exhibit the typical characteristics of MAS applications, such as multiple actors, distribution, and heterogeneity, as M4SC builds upon the MAS metaphor.

Table 1 uses ticks (✓) to indicate the presence of that (non)essential characteristic in the particular category of applications. For instance, Mirror worlds always expose the nonessential characteristics of ‘Hybrid human/software agents’ and ‘On-line/real-time’, because of their interactive nature (non-player characters, or AI characters, interacting with human characters to create believable environments).

A dash (-) indicates that the nonessential characteristics might be present in those domains, but is not necessarily there. For instance, ‘Dynamics of population’ is a characteristic that pertains to some examples of Flexible Workflows (*e.g.*, flexible service composition [5]), whereas others do not need it at all (the flexibility of the workflow follows from other aspects, the agent population remains fixed, and might be known on forehand). On the other hand, a dash might also indicate that a characteristic is not fully applicable to a particular category. For example, in Social Simulation, it is less

Characteristics	Markets	Mirror worlds	Social simulation	Policy making	Flexible workflows	Critical infra. analysis	Collective decision-making
Institution-driven	✓	✓	✓	✓	✓	✓	✓
Independent aims	✓	✓	✓	✓	✓	✓	✓
Opacity of actors	✓	✓	✓	✓	✓	✓	✓
Explicit regulation	✓	✓	✓	✓	✓	✓	✓
Complex	-		✓	✓	✓		
Hybrid human/software	✓	✓	-				-
Online/real-time	✓	✓		-	-	-	✓
Dynamics of population	✓	-			-	-	-
Revision	-	-	-	✓	-		

Table 1. Characteristics exhibited by the categories of M4SC applications.

likely that there is a mixture of agents of different types (different frameworks/reasoning mechanisms) as this might (negatively) impact the simulation results. However, a mixture of software and human agents might be possible.

The absence of a tick or dash, naturally, means that that characteristic is very unlikely to occur in that particular category of application. For example, in the case of Critical Infrastructure, it is unlikely, or even unwanted, that there are mixed agents. The similarity of the agents, *i.e.*, the fact that they are all software agents and all build using the same framework, is essential to the correct working of Critical Infrastructure systems. Using mixed agents could lead to difficulties and unforeseen situations that are, in this category of domain, rather avoided.

The remainder of this section comprises one page summaries of substantial examples, that have been implemented using the various frameworks described in this volume, to give an indication of the application of M4SC within each of the different categories.

B2B E-Contracting[5]

Category: Markets. **Framework:** ANTE.

Goal: Provide automated tools for agent-based electronic contracting in business-to-business (B2B) scenarios, taking advantage of an integrated approach based on negotiation, contractual norm monitoring and computational trust.

[5] This exemplar was written by Henrique Lopes Cardoso, University of Porto, Portugal.

Lessons learned: The integration of automated negotiation, contract monitoring and computational trust is a sensible direction to approach B2B contracting with multi-agent technology. While delegating negotiation decisions to software agents might be a daring step, monitoring contracts and building trust on past evidence comprise tasks that are more easily accepted.

Time line: The B2B E-contracting study lasted for three years. Different models have been integrated throughout the project's lifetime, taking input from several SMEs.

Description: Commercial relationships between business partners in the digital economy are increasing in flexibility, and business deals tend to be created whenever a business opportunity arises. Moreover, the instability in demand increases the need for enterprises to search for new partners, with an associated risk of dealing with entities whose previous performance might be unknown beforehand. Therefore, enterprises need mechanisms that allow, not only to evaluate the confidence they have on current or potential partners (and to monitor this confidence in a continuous and automatic way), but also to prevent deceptive behaviours from partners or assist on reacting upon them when prevention is not possible. In pursuing such a desideratum, several research communities (*e.g.*, social sciences, psychology, economics, and distributed artificial intelligence) are studying and proposing approaches for the development of trust and reputation models, to be applied in the automation of the partner selection process. Also, research is being done on the use of these models in the areas of contract establishment and enforcement.

In this example case, we aimed at studying and developing models that concern the relationship between trust and normative environments, with the purpose of applying them to e-contracting negotiations. The ANTE platform [65] addresses, in an embracing way, the issue of electronic contracting. Negotiation is used to select, from a group of potential partners, the best ones to meet a concrete business opportunity. Contracts resulting from successful negotiations are validated, registered and electronically signed. Such contracts are then passed to the normative environment for monitoring purposes. Finally, the contract performance of agents is assessed and stored for building metrics on their reliability. A repository containing this information enables us to close the cycle by exploiting its use in further negotiations.

As a proof of concept, the idea behind using ANTE in a B2B e-contracting scenario [64] is to allow SMEs to be represented in a virtual market by software agents, capable of semi-automatically entering in negotiations. In a first approach, the idea was to provide predefined contract types whose normative content could be monitored. Furthermore, agents have been configured with specific contract performance profiles (based on probabilities, handicaps, or more sophisticated models), allowing the collection of historical data that would feed trust models. The evolution of this ecosystem is then observable by making use of such trust models when selecting negotiation partners.

mWater[6]

Category: Markets. **Framework:** ROMAS-MAGENTIX2.

[6] This exemplar was written by María Emilia García, University of Valencia, Spain.

Goal: Implement a market for water rights, including the model and simulation of the water rights market itself, the basin, users, protocols, norms and grievance situations. This application was used as a testbed for agreement technologies.

Lessons learned: There have been several key points where the use of a regulated open MAS design methodology and executing platform have been shown as crucial: (1) Markets are open environments where entities can enter and leave the system during runtime, (2) It is necessary to integrate existing (running) systems that can also change during runtime, (3) The specification of norms and contracts between entities and organisations clarifies the design and ensures the stability of the implementation.

Time line: The development of this case study was part of the Agreement technology project that lasted five years.

Description: The case study was designed using the ROMAS methodology and implemented using the MAGENTIX2 platform [37]. The water market is regulated by the National Hydrological Plan of the country that establishes the creation of one *basin institution* for each water basin. Each basin institution is implemented as an autonomous organisation. In addition to the National Hydrological Plan, these institutions must follow the specific regulations of their region. These regulations have been implemented as specific norms of the Magentix2 platform. Each basin institution offers a set of software services and resources to its members. These applications have been developed by different developers using different technology. In order to allow the interoperability of the system the interchange of services and products have been specified using contracts and web services.

Different autonomous entities, representing individuals, groups of irrigators, industries, or other water users, get in contact in order to buy and sell water rights. They are able to negotiate the terms and conditions of the transfer agreement following the specific regulations of the basins involved. Each entity of the system was implemented as an agent and each group of users as an organisation.

The case study was also implemented using the Electronic Institutions (EI) framework [69]. It allowed us to analyse the differences between the two platforms. In essence the main differences are related to the conception of organisation and the specification of suborganisations. In Magentix2 organisations are explicitly represented and this allows the specification of different normative environments.

Virtual environments[7]

Category: Mirror worlds, serious gaming, augmented reality. **Framework:** Inst*AL*.

Goal: Norm-influenced behaviour for agents controlling entities in virtual environments, with the aims of (i) enhanced plausibility when interacting with human-controlled avatars (e.g. Second Life) [59] (ii) provision of populations of non-player characters [58] and (iii) exploration of coordinating behaviour in complex situations (e.g. traffic simulation in SUMO) [15].

Lessons learned: External norms in combination with an agent architecture that is capable of incorporating detached norms (i.e., obligations and permissions) into its rea-

[7] This exemplar was written by Julian Padget, University of Bath, United Kingdom.

soning [60] provide a flexible mechanism for directing agent behaviour without post-deployment reprogramming and account for behaviours that were not known at the time the agent was designed. This flexibility is however inevitably limited to behaviours that can be composed from the repertoire of actions of the agent.

Time line: The work formed the central part of two doctoral theses [14,58] and took place over a period of approximately 5 years. Conventional testing is limited to individual components while integrated testing and evaluation is carried out through scenarios, which are captured as video [13].

Description: Virtual environments and complex simulations in practice present very similar scenarios for the intelligent agent, in that the agent needs to be able to adapt its behaviour to a range of situations, but it is impractical and unrealistic to embed that capacity in the agent when it is designed, especially since new or unexpected situations can arise later. To address this problem, we use the concept of an external, explicit representation of norms (which we call an institution), in which norms (obligations, permissions) are detached (instantiated) in response to agent actions and delivered to an agent as percepts that can be incorporated into their belief base and hence into agents's decision-making. Thus in the Second Life scenario, the agents are provided by the Jason [24] agent platform and the BDI interpreter is extended to reason about norms and priorities [60], enabling the agent to switch between conventional politeness – affecting inter-personal distance and group movement behaviour, for example – when acting as a tour guide, to directing visitors when an emergency occurs – when moving with a given tour group and not mixing with other tour groups is no longer appropriate. The SUMO simulation scenario is enriched by the use of multiple interacting institutions, that have different but related competencies, to handle a road traffic accident in the slow lane of a three-lane motorway [13], namely (i) the group of vehicles involved in the crash (ii) the emergency services (iii) the insurance companies of the group of vehicles, and (iv) motorway management, which handles activities such as lane merging. In this particular scenario, cooperative behaviour with respect to lane merging, so that traffic in the same lane as the accident is able to move out and past, leads to higher overall average speeds (*i.e.*, a collective benefit). The Second Life and SUMO scenarios make use of the Bath Sensor Framework[8] to provide a lightweight scalable communications framework, using a topic-based publish-subscribe model, to connect the various components, while the institutional models are specified in Inst*AL* [71].

Platform Independent Simulation of an Altruistic Bats Society[9]

Category: (Social) simulation. **Framework:** INGENIAS.

Goal: Performing agent based simulation in several simulation platforms using one same INGENIAS model. Using the approach to reproduce a known case study about altruism in a society of bats.

Lessons learned: Transforming a specification into any kind of product, such as problem description for a social simulation framework, can be done. However, it requires a

[8] https://github.com/mas-at-bath/bsf

[9] This exemplar was written by Jorge Gómez-Sanz, University of Madrid, Spain.

significant amount of work to find out which information is really needed at the modeling level and how these models have to be translated into platform-specific code. An agent-based model can express concerns related with social simulation problems by using organizational concepts, but the inclusion of platform-specific information makes the modeling task harder for non-experienced users. The experiment confirmed that a smart bats society will choose martyrdom as a successful strategy when altruism is involved.

Time line: It was developed within the INGENIAS 2 project in 3-4 months by a team of three part-time members.

Description: The relevant literature introduces several tools for designing and performing social simulation. In particular, agent-based simulation tools are increasingly popular. Current literature lists a number of them, each one with particular advantages and drawbacks. However, there is no common social simulation specification language that allows running the same social simulation problem in any of them. Since the focus is agent-based simulation, one would expect that such social simulation specification language ought to include, or be derived from, agent concepts. The case study addressed this problem and studied whether a social simulation performed with agent-based simulation tools could be specified with an agent-oriented modeling language. Using INGENIAS [43] as starting point, [75] proposed to enrich the language with primitives which extended basic INGENIAS components and to assess how well they could capture a social simulation experiment. The chosen experiment was extracted from the social simulation literature. There is a society of bats living in roosts. They hunt and return to their home. Once there, they perform social activities namely grooming and sharing food. In the modeling proposal, bats were agents and roosts were represented as agent groups. This way, bats belonging to different roosts will require modeling Intergroup helping activities. Altruism made bats prone to help each other: the more they are helped, the more they want to help; however, there is a side effect if the bats do not receive the same help in return. Bats were allowed to play different strategies. They could behave as cheater, prudent, fair, generous, or martyr. The example was modeled using INGENIAS and transformed into an executable specification for the agent based simulation tools RePast [28] and Mason [66]. Despite being effective in letting users choose their preferred simulation platform, the participation of engineers is still fundamental in this process. Several iterations were needed to improve the transformation process from an INGENIAS model to a RePast/Mason specification until it executed properly. Also, this transformation allowed the comparison of the same specification in two different simulation platforms. This led to discoveries of possible inconsistencies in the transformation process and an extensive debugging, specially of the Mason platform experiment implementation.

Policy representation, reasoning and revision[10]

Category: Policy-making. **Framework:** Inst*AL*.

[10] This exemplar was written by Julian Padget, University of Bath, United Kingdom.

Goal: To explore how formal representations of policy can be used to (i) evaluate whether a policy meets requirements (ii) identify conflicts between policies and (iii) make new policies that avoid those conflicts.

Lessons learned: Writing policies in Inst*AL* can encourage over-focus on the detail of triggering events, while it can also be difficult to find the right balance between detail and abstraction to avoid artificiality in the scenarios for evaluation. On the positive side, the capacity to use the model to verify that use-cases (specified as traces) are appropriately handled is complemented by the means to discover traces that lead to undesirable states and hence uncover potential problems with policies.

Time line: The work was in part the main theme of a doctoral thesis [61] and took place over a period of approximately 4 years. The principles developed in the thesis have been applied in a number of use cases described below and continue to be developed to address the modelling of policy hierarchies [56].

Description: The security scenario in [78] examines the assessment of the vulnerability of an asset subject to a security policy that is formalized using Inst*AL* [71] to capture the obligations on actors in an organization. By characterising several vulnerable situations (undesirable model states) and some initial conditions, sequences of actions not covered by the policy are found that lead to the loss of the asset and hence the (manual) revision of the policy to incorporate new obligations.

In practice, an actor is typically subject to the governance of several policies at the same time and those policies are often defined independently by different organizations. As a result, an actor could (unintentionally) be in an untenable situation where an action is permitted by one policy and not by another (defined as weak conflict), or obliged by one and not permitted by another (strong conflict). The first scenario [62] demonstrates the automatic conflict detection between two independent policies in which the first stipulates an upper bound on the working hours for the holder of a student visa, while the second sets a minimum number of hours that the holder of a studentship shall work. By constructing a composite trace for the two models, the constituents of the conflicting normative states are identified and then, by using the trace as a negative example, an inductive logic programming theory revision task is able to suggest a change that resolves the conflict. The second scenario [63] considers *interacting* models, where an action in one can trigger an action in another or a state change in one may affect the state of another, as illustrated in [71]. The illustration is a data privacy case involving Facebook Ireland in 2013 regarding EU law, that prohibits the export of personal data outside the EU unless equivalent data protection applies and consent has been given, and US law that obliges Facebook to comply with a governmental data request. The analysis identifies the presence of both weak (with consent) and strong (without consent) conflicts.

What these examples demonstrate is that for suitable levels of abstraction, normative models of (in this case) security or privacy policies can be examined and evaluated at the design stage to identify issues for policy-makers' attention. Furthermore, those same models are subsequently deployable for the purpose of compliance monitoring.

Machine-to-Machine Governance System[11]

Category: Flexible workflows. **Framework:** JaCaMo.

Goal: To provide an adaptive governance framework to coordinate stakeholders' business strategies and domain-specific information processes on top of a Machine-to-Machine (M2M) infrastructure. Besides scalability management issues, the aim is to also provide an efficient response to infrastructure dynamics.

Lessons learned: Decentralized and autonomic adaptation of the governance process, using both self-organization and reorganization techniques, is a suitable approach to deal with the distributed, heterogeneous, constrained and large scale nature of M2M infrastructures. The JaCaMo first class abstractions [23,22] for Agent, Environment, Interaction and Organization facilitate the explicit representation of governance levels and to set them in synergie by a self-organization and reorganization process.

Time line: The project consisted of a study of the governance dimensions in M2M Systems in the context of Smart Cities and the implementation of a demonstrator of the governance process on a simulated M2M system. The project lasted three years (see [77] for details).

Description: M2M is a key technology in city-scale deployments of networks interconnecting together web application services and real-world sensors and actuators. We focus on how the programming dimensions of JaCaMo have been used to define the social coordination required to govern such M2M infrastructures. The governance strategy level has been represented through a *horizontal* organization – built from ETSI TC M2M standard recommendations for global management of M2M infrastructures – and *vertical* ones – each expressing a stakeholder collective strategy and SLA. Such a separation among organizations reduces the complexity of their management and adaptation. Each organization defines roles, *i.e.*, capabilities in the M2M infrastructure, social schemes, *i.e.*, functionalities structured into goal trees, and norms stating which goals to achieve by agents when playing some roles.

The binding of the horizontal and vertical organizations is done through agents playing roles in both kinds of organizations. Agents implement the governance tactical process by realizing the strategic objectives issued from the organization, finding trade-offs or adaptations of the organization when no other action is possible. Each agent governs at most one M2M entity (a client application, a core platform server, or a gateway of a group of devices) of the infrastructure and adopts one or several roles in the organization corresponding to its responsibilities in the governance. Agents enact the governance in the management of the M2M infrastructure but also refine and adapt the horizontal and vertical governance in situations where achievement of goals appear impossible or not efficient enough.

Governance artifacts in the environment abstract and encapsulate the governed M2M entities under the control of agents. Each artifact monitors an M2M entity using infrastructure perception modules and notifies the agents about the current state of the M2M infrastructure through governance properties, for performance statistics, and events about critical states for quick governance decision. Agents can then control the M2M entity using governance operations that can trigger actions directly on the M2M in-

[11] This exemplar was written by Olivier Boissier, EMSE, France.

frastructure by means of infrastructure controllers. Governance operations can perform more complex processes, such as set up and tuning of the monitoring components or the governance policies that each artifact embeds.

Tourism service[12]

Category: Flexible workflows. **Framework:** ROMAS-MAGENTIX2.

Goal: Implement an application that offers tourism services to different users of a city including the capability to plan different activities in a given day according to their preferences. The main goal of this application is to provide tourism services to users in a coordinated way.

Lessons learned: The integration of different technologies as web services, multi-agent systems and planning algorithms into mobile devices is challenging but with immense benefits as it facilitates the interoperability and versatility of systems.

Time line: This project lasted for one year.

Description: Starting off with the need to develop an application focused on tourism information services of different types to users; and intending to take advantage of recent technological advances in mobile/smart devices capable of connecting to the Internet, a MAS has been developed using the Magentix2 platform [37].

The system is basically formed by three kinds of agents: the BrokerAgent, the SightAgent and the UserAgent [88]. The BrokerAgent will provide the services to the UserAgent and has updated information of the diferent places of interest existing in the city of Valencia. This agent is in charge of establishing a communication between the UserAgents and the different SightAgents. The SightAgents contains all the information about the different places of interest in a city. Each SightAgent stores all the necessary information corresponding to one place of interest. Finally, the UserAgent gives the user (tourist), the possibility to interact with the system by means of a friendly GUI running on a mobile device. The BrokerAgent will receive requests and actions from the different UserAgents, and it will send them to the corresponding SightAgents. The SightAgents will compare the received information with his own, and, if this information matches, the SightAgent will send some information about himself to the BrokerAgent. Then the BrokerAgent will send this information to the UserAgent.

The implemented system use the TurOnt ontology, created using Protégé. This ontology gives a detailed description of the different kind of places that are relevant to tourists. Moreover, this ontology gives actions and predicates that help agents to communicate with each other and that allows running planning algorithms to adapt the offer to the user's preferences.

PeerLearn: An environment for running flexible on-line lessons.[13]

Category: Flexible workflows. **Framework:** EI/EIDE.

[12] This exemplar was written by María Emilia García, University of Valencia, Spain.
[13] This exemplar was written by Pablo Noriega, IIIA-CSIC, Spain.

Goal: Provide high school or college instructors with the means to define and execute on-line lessons in a flexible way.

Lessons learned: There are several tools for on-line education that support teachers in the management of lesson plans on the web. Few are task-oriented and support any form of their on-line "execution". We found that the use of electronic institutions is a convenient approach to meet those needs.

More specifically, (i) An electronic institution provides the on-line coordination support at the same time that it establishes and enforces terminological, interaction and procedural conventions that permeate the system. (ii) One can build a re-usable general-purpose lesson management system around variations of electronic institutions with a common class of scene patterns, ad-hoc web services, software agents and repositories. (iii) SIMPLE, a trimmed-down version of the ISLANDER specification language[14], allows for quick specification and debugging of a large class of electronic institutions. (iv) The proficient use of that simplified graphic interface requires no IT background from users and almost no training.

Time line: The development of *PeerLearn* answered a request of high school teachers, who found *PRAISE* ideas worth taking up for their everyday teaching. The first version was liberated in six months. It is being used (since Sep. 2015) in a beta-test mode for running actual courses (English and music playing) in the Barcelona area.[15]

Description: *PeerLearn* is a general-purpose on-line lesson support platform with four main modules: *PeerAssess* where the teacher/pedagogue define(s) the evaluation rubric of the lesson; *LessonEditor* used by teachers to define a lesson plan (specify an electronic institution); *PeerFlow* to execute the lesson plan, and *ColAssess* to evaluate the students' performance.

PeerLearn is built on the following design premises: (i) Users are high school or college instructors and students with no IT expertise. (ii) Instructors use the platform to specify and run their on-line course by themselves. (iii) A course is composed of "lessons" where students are intended to complete certain "tasks". (iv) Lessons may be organised in non-trivial flows of activities (repetitive, concurrent, open) that involve role-dependent interactions among individuals and groups of individuals. (v) Activities may involve the use of services like automatic evaluation, execution of practice exams or access to external advisors. (vi) Documents may be text, image, sound or video. (vii) Specification of a lesson is made with a simple graphical interface. (viii) Lessons are compiled and executed as an electronic institution using the EIDE framework. (ix) The course runs within a platform that takes care of user identities, access to services and repositories, and the handling of documents

The salient virtue of the system is to have a powerful way to specify non-trivial lesson plans that can then be executed on line by a group of students who are engaged in individual and in collaborative tasks. Lessons are specified with SIMPLE, which is essentially a nicer and simpler version of the EIDE specification language, ISLANDER. It is less expressive than ISLANDER (in as much it does not allow ubiquity of agents,

---

[14] See [69] for details about EI/EIDE and the include ISLANDER specification language.

[15] *PRAISE* was a three-year (Oct'12 - Sep'15) FP7 project, see http://www.iiia.csic.es/praise, that produced a working prototype for collaborative music education that became the basis for *PeerLearn*, and motivated the SIMPLE [31] graphical language.

run-time creation of new scenes, or the more complex transitions between scenes), but because of this reduced expressiveness the graphic conventions are more intuitive: a palette of icons and templates, and point-drag-click actions are used to establish the performative structure. With this palette, the instructor can specify roles, choose from a repository of available scenes – and define new ones – that correspond to learning tasks, state how individuals playing a particular role may move from one scene to others, specify the “documents” that are involved in each task, and also make use of relevant services and ad-hoc software agents. One such service, for example, is an automated “music critic’ that evaluates the quality of an actual performance by a student. This critic compares the new performance against a teacher’s performance and against the recordings of previous executions by the same student to identify the segments that need work and those where proficiency is already achieved.

Warehouse management control systems[16]

Category: Flexible workflows. **Framework:** OperA.

Goal: Implement a control system that performs about as efficient as the Operations Research solution (centralised/optimised solution), but which provides more modularity, flexibility and robustness [3].

Lessons learned: Social coordination frameworks provide a useful metaphor for the design and implementation of complex, distributed planning systems. Even though no agents or organisation may appear in the actual code, thinking of the domain in terms of agents, organisations and associated control structures, helps clarify design assumptions and choices [52].

Time line: The research of the Falcon project[17] comprises a pilot study and implementation of a simulation framework to validate the results. The project lasted one year.

Description: Warehouse management and control systems (WCMS) are traditionally optimised (through Operations Research) to a specific situation and do not provide the flexibility required in contemporary business environments. The hardware that is used in warehouses has been subject to evolution which resulted in an increasingly modularised approach to decomposing the warehouse tasks and improving reusability. This evolution is, however, not reflected in the software that controls these machines and many WCMSs are still centralised and monolithic. Recently, multiagent systems were proposed to introduce more flexibility in warehouse control. The decentralised nature of autonomous agents improves the modularity of a WCMS over a centralised approach. Moreover, the decentralisation, combined with standardised interfaces between the agents, can lead to an improvement of reusability and exchangeability. However, it also decreases the system’s performance and predictability; the agent system is harder to optimise to specific situations, and the autonomy of the agents, which leads to better flexibility, also makes it harder to predict system behaviour. The main problem here is that aspects such as efficiency, flexibility, and robustness are aspects that pertain to the system as a whole. One cannot optimise all three at the same time, but one has to

[16] This exemplar was written by Huib Aldewereld, Delft University of Technology, NL.
[17] http://redesign.esi.nl/research/applied-research/finished-projects/falcon/

balance the three aspects. Due to the distributed nature of the agent system, it is harder to guarantee this overall balance.

We applied social coordination mechanisms to regulate the agents within the control system. The organisation specifies those aspects of the system that need to be guaranteed by the agents together. The resulting system (see [4] for more details) splits the domain into three 'levels' (called spaces): 1) a planning space, where the agents interact to assign orders and reserve stock for order fulfilment, 2) a scheduling space, where the agents interact about the scheduling of crates containing stock or orders, and 3) the plant (execution) space, which is the physical dimension where the hardware components interact (i.e., conveyor belts, automated storage racks, and (automated) picking stations). Agents can have a presence in each of these spaces (but some agents are limited to some of the spaces). Each of the spaces (and thus the presences in that space) are constrained; the plant space is constrained by the physical layout of the plant (*e.g.*, only components that are physically connected can interact), whereas the other two spaces are governed by an agent organisation (designed in OperA [2]). The organisation specifies the responsibilities and rights of the different components on that level, and governs how the components can interact with respect to planning and scheduling.

Decentralized and distributed Multi-Agent Systems for Smartgrids[18]

Category: Critical Infrastructure. **Framework:** INGENIAS.

Goal: Address the way software agents are being used in the domain of Smartgrids and propose new coordination mechanisms.

Lessons learned: Most agent-oriented solutions found in the literature tend to consider agents as regular processes. Such approaches can be recognised because they identify a single control unit in charge of everything; there are restrictions in the number of agents playing some roles; and the fact that introduced agents do not show a relevant degree of autonomy. By considering peer-to-peer interactions, the flexibility of such systems grows significantly, being capable of a greater fault tolerance (there can be several agents doing similar functionality in different computation nodes), and making it easier for the Smartgrid to grow, since there are no central nodes limiting the communications.

Time line: The research was made within the three-year project MIRED-CON[19], whose goal was to produce a pilot implementation of an intelligent distributed and decentralized solution for Smartgrid control.

Description: Smartgrid control is a domain where most authors cite agent technology as a key enabler. However, there is a wide gap between those using this technology, mostly electrical engineers, and the producers of this technology, the agent researchers. For the agent researcher, there is a demanding learning process where basic alternating current concepts have to be re-learnt together with the particularities of high voltage systems. In the case of the electrical engineers, the issue is learning the subtleties of agent technology and the differences with the more widespread component- or service-based approaches. The study of both worlds was made in [47], which introduced criticism

[18] This exemplar was written by Jorge Gómez-Sanz, University of Madrid, Spain.
[19] http://grasia.fdi.ucm.es/energy/miredcon.html

on some research work introducing agent-oriented solutions. The paper illustrated with INGENIAS [43] a case of peer-to-peer interaction and how this enabled a more decentralized approach. The solution was to intensely use role-based models when designing interactions. This way, designers had to consider cases where agents interchange their roles or having more than one agent playing one of them. Another product of the project was a simulator for Smartgrids [41], the SGSimulator[20]. This simulator/emulator would enable agent researchers with little expertise in Smartgrids to actually experiment with this domain problem and propose or reuse agent coordination algorithms. One of the open issues is whether the kind of software agents researchers are used to will actually perform under the hardware constraints of a Smartgrid. The MIRED-CON project strict hardware constraints prevented the use of just any kind of conventional agent libraries. Even libraries created to work in mobile devices were deemed too heavyweight. Therefore, a very simple control solution with basic monitoring, communication, and decision capabilities was designed. The intelligence came from the opportunistic use of genetic algorithms to decide the control strategy. Hence, another lesson is that a straightforward application of agents to this domain is not trivial. Traditional agent software approaches require too much computing power to be allocated to the kind of embedded hardware being used today. That may be a reason why most researchers perform experiments by attaching a PC to the Smartgrid. In any case, the agent research can also give hints of how to solve certain problems and guide an alternative implementation, such as how to design a peer-to-peer interaction among agents and allowing them to achieve common goals.

An Agent Oriented Design and Implementation of a Delphi Process[21]

Category: Collective Decision Making. **Framework:** INGENIAS.

Goal: Evaluating the Delphi process, a collective discussion process used by humans, when performed by agents within the context of the document classification problem.

Lessons learned: A protocol involving human parties can be realised in terms of agents. However, elaborating the content of the discussion is the most challenging part. For particular domains, it is possible to automatically understand the content and to mimic the mechanism used by humans to generate answers and new questions. However, other domains are not that well suited because understanding the contents of the discussion is an open problem. The design and implementation of the solution were carried out with INGENIAS [43].

Time line: This case study was developed by a four-person team in four months with nearly full time dedication. It is work done within the INGENIAS 2 and SociAAL projects.

Description: The Delphi process is well-known in social sciences. It is used to organize discussions among experts in order to achieve meaningful conclusions. The process is moderated by one individual, who receives an initial question which is forwarded to each expert. Answers to the questions are collected and analysed. If experts do not

[20] http://sgsimulator.sf.net

[21] This exemplar was written by Jorge Gómez-Sanz, University of Madrid, Spain.

agree, a new round is started with a rephrase of the initial question to advance the discussion. These principles were modeled with INGENIAS [43] and implemented automatically with its code generation facilities. The chosen domain for testing the implementation was a document classification problem [38]. Given a plain text document, agents had to discuss and decide whether this plain text document should be accepted or not as part of the collection. The system defined several roles: a client that creates the input documents, a moderator that drives the experts's discussion, and the experts, who decide to what extent a document should be accepted. Agents playing the role of experts were initialized with selected pieces of documents. Documents were pages of Wikipedia. Each expert agent had the capability of comparing an external document with its documents and return an evaluation of its significance in terms of found keywords. The moderator used this evaluation to rephrase the question by incorporating the keywords each expert found. So, in each round, the expert agent evaluates a slightly different document that contains feedback from other experts from the previous round. The experiment addressed some relevant scenarios, such as documents rejected by some experts and which were eventually accepted because of the continuous modification of the original question; the opposite also happened. The scenarios explored used human evaluated documents, giving excellent precision and recall scores. Nevertheless, there was a computation overhead in the process that prevented an extended application of this technique for document classification on a higher scale. The ideas were revised in [46] in an attempt to revisit INGENIAS modeling primitives and convert the old version of the protocol onto a more reusable one. The new specification extensively used the concept of a workflow, which highlights tasks and their performers whereas previous work [38] focused on protocols. The specification was apparently simpler. It had fewer diagrams and concepts, which shows that INGENIAS gained expressive power. Also, the new version identified the required parameterization of a new version of the Delphi protocol. This parameterization would facilitate the reuse of the Delphi protocol in different suitable domains.

4 Framework examples

Given the categories presented previously, we now present a brief overview of other examples of applications of social coordination by presenting application examples each of the frameworks included in this book in table 2.We also include below a short description per reference of examples (except for those pertaining to the exemplars mentioned earlier).

ANTE As mentioned in the ANTE chapter [64], the prime application that ANTE has looked into is that of B2B e-contracting or e-procurement [65].

EI/EIDE

[83] Fishmarket. Traditional fish auctions; this is the earliest application (1996) and the inspiring example of what became the EI/EIDE framework. Other markets have also been implemented in the EI/EIDE framework, for example:

Framework	Markets	Mirror worlds	Social simulation	Policy making	Flexible workflows	Critical infra. analysis	Collective decision-making
ANTE	[65]						
EI/EIDE	[1],[8], [68],[83]	[21,91]	[20],[32]	[25]	[26],[27], [81,82]	[12]	[29],[57]
INGENIAS			[74,75], [86]		[48]	[47,50,51], [44],[73]	[40,49], [38,45], [42]
Inst*AL*		[15], [59]	[16], [17], [18]	[56], [62] [63]	[89]	[72], [78]	
JaCaMo		[19]		[95]	[87],[84], [76,77]	[30]	[90]
ROMAS-MAGENTIX2	[39],[34]			[36],[88]			
OperA		[53], [92,93]		[54]	[33],[55],[5], [3,4,52]	[6,80]	[67]
RTEC							[9,10,11,79]

Table 2. Examples of application by each framework.

[8] Electricity market; this example also shows how services may be attached to an electronic institution.
[68] a market-place for "open innovation"; a variant of the core design, *mWaste* is used for trading by-products of industrial processes.
[1] The ABC4MAS platform uses the EI/EIDE framework for sequential mixed auctions for supply networks.
[91] Virtual World Builder Toolkit (VWBT) is an extension of the EI/EIDE framework to design and implement 3-D virtual (electronic institutions) institutions; this paper implements the toolkit with an immersive version of a fish market, along the ideas proposed in [21].
[20] Simulation of an archaeological site as a normative multi agent system using 3D virtual institutions.
[32] Modelling of hunter-fisher-gatherer societies with a normative approach using the EI conceptual model.
[25] *mWaterDSS* is an institution for testing demand-side policies for water management. It consists of a water bank and a simulation environment for testing norms that regulate trading of water rights and their effect in the use of water in a river basin.
[26] An EI-based tool for creating on-line courses.

[27] CHARMS is an extension of the EI/EIDE framework to handle flexible specification of process models ("charters").

[81,82] are other examples of flexible workflows based on the EI framework are for hotel and for medical management systems.

[12] Proposes the use of electronic institutions to manage grid environments.

[57] Is an app of an electronic institution for mutual support communities where individuals seek and provide help for day-to-day activities.

[29] Describes WeCurate, a system to support the design of workflows for group activities, like curating a collection of images or designing a visit to a museum.

INGENIAS

[86] Cinema tickets: a case study where there are simulated users, their personal assistants, and cinemas. Users look for cinemas with certain features. The personal assistants must find a cinema satisfying the user constraints. The case study permits to evaluate information flows and check if the system dynamics match expectations.

[48] Business use case modeling: the Juul Møller Bokhande problem where a traditional library has to transform its business process and establish bonds with local universities. The case is modelled using INGENIAS.

[44] Ambient intelligence: an extension of INGENIAS to deal with ambient intelligence problems. The case study is about a teacher tracking system.

[73] The Surveillance of areas: it models with INGENIAS a surveillance system for a factory. Coordination between devices is realised through agent to agent interactions.

[40,49] Online communities: a trust & reputation system applied to filter documents relevant to communities of users. The system is modelled later on with INGENIAS in the PhD where INGENIAS was created.

[42] Business intelligence: an INGENIAS based design for a technological surveillance system. In this system, users suggest information sources to be included in the surveillance of technological advances. Each information source have to be evaluated to determine its utility. This process is expensive. The system uses a trust & reputation solution to reduce the number of evaluations. Hence, reputed persons suggesting information sources are less evaluated. The paper suggests how to perform simulations to evaluate this kind of systems.

Inst*AL*

[15] Agent controlled "self-driving" vehicles that act individually within a background population in a complex traffic simulation environment (see [14] for more detail) to demonstrate the capacity to explore policy and behaviour in a mixed human-agent driver setting.

[16] Uses the context of mobile phones and offloading cellular download by sharing cached content peer-to-peer over wifi to explore the costs and benefits of cooperation for battery life.

[17] An empirical exploration of the cost of enforcement through examination of the parameter space affecting the number and cost of enforcement agents on the level of compliance, which confirms the intuition of diminishing returns, beyond a certain point, from an increasing volume of policing.

[56] In contrast to [61], where institutions interact at the same level, develops the notion of institutions governing institutions through meta-norms that place constraints on the form of norms in governed institutions.

[59] Plausible non-player character (NPC) behaviour in Second Life (see [58] for more detail), achieved through the use of simple social institutional models that guide NPC choices.

[62] Examines the (automatic) detection of a policy conflict between two sets of regulation, where one (national) has precedence over the other (university) between legislation and the revision of the latter to comply with the former.

[63] Builds on [62] to analyse an interacting legislation scenario inspired by the conflict reported in 2013 between the EU Data Directive on privacy and the requirements imposed on US-registered companies by the National Security Agency (NSA) in respect of Facebook in Ireland.

[72] Analyses a model policy concerning the privacy and transparency of personal data to uncover when the data becomes vulnerable and later compromised.

[78] Evaluation and empirical revision of a formal model of security policy that examines the effectiveness of obligations on employees in protecting assets.

[89] Uses the setting of a scene from a play, performed by puppets controlled by BDI agents (augmented with a simple emotional model), to illustrate the scope for both interactive and generative narrative, where the narrative structure is represented as a normative framework.

JaCaMo

[87] Smart Building Management: use of the JaCaMo platform to support ambient computing applications used to manage the use of a smart building.

[19] Intelligent Virtual Environment (MAM5): use of the JaCaMo platform in cooperation with a gaming platform to support coordination and distributed games involving multiple actors.

[95] Trust management in Virtual Community: use of the JaCaMo platform to implement an open innovation management tool in which users can define trust management policies dealing with the sharing of documents and ideas in such a platform.

[90] Knowledge Management: use of the JaCaMo platform to define a knowledge management system.

[84] Automation of small series production: use of the JaCaMo platform to deal with the complexity of the implementation of small series production lines. Small series requires high abstract constructors to simplify its adaptation to new products.

[30] Crisis management: definition of a socio-technical system combining tangible surfaces with the JaCaMo platform in synergy with Situated Artificial Institution Model to coordinate human actors participating to different organizations in the management of crisis.

ROMAS-MAGENTIX2

[34] Use of the ROMAS methodology and Magentix framework for technical assistance and virtual markets.
[35] Application of the ROMAS methodology to academic domain.
[36] Analysis of agent-oriented engineering, and using organisational elements, in the modelling of e-Health systems.

OperA

[33] Support for systems for knowledge management that incorporate the management of knowledge assets with the facilitation and encourgement of interaction between people in an open environment.
[67] Models for scenario development to develop and evaluate organizational/corporate strategy.
[55] Analysis and design of inter- and intra-organizational interaction in the maintenance systems of the Dutch railways.
[6,80] Simulations of crisis scenarios for the purpose of the evaluation of critical infrastructure and control mechanisms.
[54] Process and regulation compliance verification for Customs authorities in International Container Trade.
[92,93] Formalisation and implementation of adaptive serious games using multi-agent organizations.
[53] Formalization of improvisation theatrical performances (formalising dynamic interplay between actors on a high level of abstraction).
[5] Dynamic, flexible service compositions using organisational models to provide the context (aims, objectives, vision) for the system.

RTEC

[10] A theoretical and computational framework for the executable specification of heterogeneous and open systems (characterised by limited trust and unpredictable behaviour).
[79] A characterisation and specification of a general voting protocol, an essential element of mechanism design, to ensure robustness by respecting the way in which the votes are cast and the outcome is declared.
[9] Application of framework for open agent systems to ad hoc networks, where computational systems whose members may fail to, or choose not to, comply with the rules governing their behaviour.
[11] Application of the specification framework for open systems to the execution of a contract-net protocol.

5 Summary

It should not come as a surprise that M4SC share all of the features that differentiate MAS from conventional software applications. The purpose of this chapter is to illustrate the additional, and specifically, what we regard to be the essential characteristics, that further distinguish M4SC. First and foremost is the adoption of institution-driven design, meaning that the central focus for the expression of system requirements stems from the capture, representation and reasoning for the norms that will guide participant behaviour. The independence and opacity of actors arises naturally from this separation of concerns, in which regulator and participant are distinct and neither makes assumption about the other: their common point of reference is the fourth essential characteristic, the explicit representation about which participants (human and software) can reason and from which they can make informed decisions.

To provide concrete illustration of the essential characteristics, we discuss a range of application domains that we consider feature (or have the potential to need) a greater focus on coordination and governance, a separation of design concerns between environment and participant and scope for the use of institutions as repositories and sources of (system) regulation. In consequence, we believe M4SC can deliver the desired levels of trust, confidence and flexibility in coordination and control that are hallmarks of physical-world social institutions. Finally, we use a variety of case studies, developed using the frameworks in this volume, to show progress towards these goals and to show what remains to be done.

M4SC is a work-in-progress. Open issues range from the user-friendliness of the modelling approaches to dealing with real-time and concurrent environments, from verification to semantic interoperability, and from uptake by industry to the use of M4SC for Responsible Innovation. In the next chapter we discuss the challenges for M4SC.

Acknowledgements

The authors would like to thank Henrique Cardoso Lopes, María Emilia García, Jorge Gómez-Sanz, Olivier Bossier, Jomi Hübner, and Alessandro Ricci for their contribution(s) to the Exemplars section of this chapter.

References

1. Alba, T. P., Mikhaylov, B., Pujol-Gonzalez, M., Rosell, B., Cerquides, J., Rodríguez-Aguilar, J. A., Esteva, M., Fabregues, A., Madrenas, J., Sierra, C., Carrascosa, C., Julian, V., Rodrigo, M., and Vassirani, M. An environment to build and track agent-based business collaboration. In Ossowski, S., editor, *Agreement Technologies (Law, Governance and Technology Series)*, volume 8, chapter 36, pages 611–624. Springer, 2013.
2. Aldewereld, H., Álvarez-Napagao, S., Dignum, V., Jiang, J., Vasconcelos, W., and Vázquez-Salceda, J. OperA/ALIVE/OperettA. In *This volume*. 2016.
3. Aldewereld, H., Dignum, F., and Hiel, M. Re-organization in warehouse management systems. In *Proceedings of the IJCAI 2011 workshop on artificial intelligence and logistics (AILog-2011)*, pages 67–72, 2011.
4. Aldewereld, H., Dignum, F., and Hiel, M. Decentralised warehouse control through agent organisations. In *Automation in Warehouse Development*, pages 33–44. Springer, 2012.
5. Aldewereld, H., Padget, J., Vasconcelos, W., Vázquez-Salceda, J., Sergeant, P., and Staikopoulos, A. Adaptable, organization-aware, service-oriented computing. *Intelligent Systems, IEEE*, 25(4):26–35, 2010.

6. Aldewereld, H., Tranier, J., Dignum, F., and Dignum, V. Agent-based crisis management. In *Collaborative Agents-Research and Development*, pages 31–43. Springer, 2011.
7. Arcos, J. L., Noriega, P., Rodríguez-Aguilar, J. A., and Sierra, C. E4mas through electronic institutions. In *Environments for multi-agent systems III*, pages 184–202. Springer, 2007.
8. Arcos, J. L., Noriega, P., Rodriguez-Aguilar, J. A., and Sierra, C. E4mas through electronic institutions. In Weyns, D., Parunak, H., and Michel, F., 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.
9. Artikis, A., Kamara, L., Pitt, J., and Sergot, M. J. A protocol for resource sharing in norm-governed ad hoc networks. In *Declarative Agent Languages and Technologies (DALT) II, Revised Selected Papers*, volume 3476 of *LNAI*, pages 221–238, 2005.
10. Artikis, A., Sergot, M. J., and Pitt, J. An executable specification of a formal argumentation protocol. *Artif. Intell.*, 171(10-15):776–804, 2007.
11. Artikis, A., Sergot, M. J., and Pitt, J. V. Specifying norm-governed computational societies. *ACM Trans. Comput. Log.*, 10(1), 2009.
12. Ashri, R., Payne, T. R., Luck, M., Surridge, M., Sierra, C., Aguilar, J. A. R., and Noriega, P. Using electronic institutions to secure grid environments. In Klusch, M., Rovatsos, M., and Payne, T. R., editors, *Cooperative Information Agents X: 10th International Workshop, CIA 2006 Edinburgh, UK, September 11-13, 2006 Proceedings*, pages 461–475. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
13. Baines, V. Challenges in artificial socio-cognitive systems: A study based on intelligent vehicles: Dataset. http://doi.org/10.15125/BATH-00080. Simulation videos.
14. Baines, V. *Challenges in artificial socio-cognitive systems: A study based on intelligent vehicles*. PhD thesis, University of Bath, September 2014. http://opus.bath.ac.uk/46550/, retrieved 2015-10-07.
15. Baines, V. and Padget, J. A situational awareness approach to intelligent vehicle agents. In Behrisch, M. and Weber, M., editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 77–103. Springer International Publishing, 2015.
16. Balke, T., De Vos, M., and Padget, J. Analysing energy-incentivized cooperation in next generation mobile networks using normative frameworks and an agent-based simulation. *Future Generation Computer Systems*, 27(8):1092–1102, 2011.
17. Balke, T., De Vos, M., and Padget, J. I-ABM: combining institutional frameworks and agent-based modelling for the design of enforcement policies. *Artificial Intelligence and Law*, pages 371–398, 2013.
18. Balke, T., De Vos, M., and Padget, J. A. Evaluating the cost of enforcement by agent-based simulation: A wireless mobile grid example. In *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, volume 8291 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2013.
19. Barella, A., Ricci, A., Boissier, O., and Carrascosa, C. MAM5: Multi-Agent Model For Intelligent Virtual Environments. In *10th European Workshop on Multi-Agent Systems (EUMAS 2012)*, pages 16–30, 2012.
20. Bogdanovych, A., Rodríguez, J. A., Simoff, S., Cohen, A., and Sierra, C. Developing virtual heritage applications as normative multiagent systems. In *Proceedings of the 10th International Conference on Agent-oriented Software Engineering*, AOSE'10, pages 140–154, Berlin, Heidelberg, 2011. Springer-Verlag.
21. Bogdanovych, A., Simoff, S., and Esteva, M. Virtual institutions: Normative environments facilitating imitation learning in virtual agents. In Prendinger, H., Lester, J., and Ishizuka, M., editors, *Intelligent Virtual Agents: 8th International Conference, IVA 2008, Tokyo, Japan, September 1-3, 2008. Proceedings*, pages 456–464. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

22. Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. Multi-agent oriented programming with JaCaMo. *Sci. Comput. Program.*, 78(6):747–761, 2013.
23. Boissier, O., Hübner, J. F., , and Ricci, A. The JaCaMo Framework. In *This volume*. 2016.
24. Bordini, R., Hübner, J., and Wooldridge, M. *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
25. Botti, V., Garrido, A., Giret, A., and Noriega, P. The role of mas as a decision support tool in a water-rights market. In Dechesne, F., Hattori, H., ter Mors, A., Such, J., Weyns, D., and Dignum, F., editors, *Advanced Agent Technology*, volume 7068 of *Lecture Notes in Computer Science*, pages 35–49. Springer, Berlin / Heidelberg, 2012.
26. Brito, I., Gutierrez, P., Hazelden, K., de Jonge, D., Lemus, L., Osman, N., Rosell, B., Sierra, C., and Roig, C. Collaborative peer assessment using peerlearn. In Steels, L., editor, *Music Learning with Massive Open Online Courses (MOOCs)*, volume 6, chapter 10, pages 145–174. IOS Press, 2015.
27. Brito, I., Osman, N., Sabater-Mir, J., and Sierra, C. Charms: A charter management system. automating the integration of electronic institutions and humans. *Appl. Artif. Intell.*, 26(4):306–330, April 2012.
28. Collier, N. Repast: An extensible framework for agent simulation. *The University of Chicago's Social Science Research*, 36:2003, 2003.
29. Confalonieri, R., Yee-King, M., Hazelden, K., d'Inverno, M., de Jonge, D., Osman, N., Sierra, C., Agmoud, L., and Prade, H. Engineering multiuser museum interactives for shared cultural experiences. *Engineering Applications of Artificial Intelligence*, 46, Part A:180 – 195, 2015.
30. De Brito, M., Thevin, L., Garbay, C., Boissier, O., and Hübner, J. F. Situated artificial institution to support advanced regulation in the field of crisis management. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection*, volume 9086 of *Lecture Notes in Computer Science*, pages 66–79. Springer, 2015.
31. de Jonge, D. and Sierra, C. SIMPLE: a language for the specification of protocols, similar to natural language. In *Proceedings of the XIXth International Workshop on Coordination, Organizations, Institutions and Norms in Multiagent Systems (COIN@AAMAS2015)*. 2015.
32. de la Cruz, D., Estévez, J., Noriega, P., Pérez, M., Piqué, R., Sabater-Mir, J., Vila, A., and Villatoro, D. Norms in h-f-g societies. grounds for agent-based social simulation. In Francisco Contreras, F. J. M., editor, *CAA'2010 Fusion of Cultures. Proceedings of the 38th Conference on Computer Applications and Quantitative Methods in Archaeology*, Granada, 06/04/2010 9998. CAA, CAA. There is a Spanish version in Cuadernos de Prehistoria y Arqueología, No. 20; pp 149-161., 2010 (U. Granada).
33. Dignum, V. *A Model for Organizational Interaction: based on Agents, founded in Logic*. SIKS Dissertation Series 2004-1. Utrecht University, 2004.
34. García, E., Giret, A., and Botti, V. A model-driven case tool for developing and verifying regulated open mas. *Science of Computer Programming*, 78(6):695–704, 2013.
35. García, E., Giret, A., and Botti, V. ROMAS methodology. In *Regulated Open Multi-Agent Systems (ROMAS)*, pages 51–95. Springer, 2015.
36. García, E., Tyson, G., Miles, S., Luck, M., Taweel, A., Staa, T., and Delaney, B. An analysis of agent-oriented engineering of e-health systems. In *Agent-Oriented Software Engineering (AOSE) XIII, Revised Selected Papers*, volume 7852 of *Lecture Notes in Computer Science*, pages 134–150. Springer, 2013.
37. García, E., Valero, S., and Giret, A. ROMAS-MAGENTIX2. In *This volume*. 2016.
38. García-Magariño, I., Gómez-Sanz, J. J., and Pérez-Agüera, J. R. A complete-computerised Delphi process with a multi-agent system. In Hindriks, K. V., Pokahr, A., and Sardiña, S., editors, *Programming Multi-Agent Systems, 6th International Workshop, ProMAS 2008, Estoril, Portugal, May 13, 2008. Revised Invited and Selected Papers*, volume 5442 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2008.

39. Garrido, A., Giret, A., Botti, V., and Noriega, P. mwater, a case study for modeling virtual markets. In Ossowski, S., editor, *Agreement Technologies*, volume 8 of *Law, Governance and Technology Series*, pages 565–582. Springer Netherlands, 2013.
40. Gómez-Sanz, J. J. Modelado de sistemas multi-agente. *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial*, (17):97–99, 2002.
41. Gómez-Sanz, J. J., Cuartero-Soler, N., and Garcia-Rodriguez, S. A testbed for agent oriented smart grid implementation. In Baldoni, M., Baresi, L., and Dastani, M., editors, *Engineering Multi-Agent Systems*, volume 9318 of *LNAI*, pages 92–108. Springer, 2015.
42. Gómez-Sanz, J. J., Fernández, C. R., and Arroyo, J. Model driven development and simulations with the INGENIAS agent framework. *Simulation Modelling Practice and Theory*, 18(10):1468–1482, 2010.
43. Gómez-Sanz, J. J. and Fernández, R. F. INGENIAS. In *This volume*. 2016.
44. Gómez-Sanz, J. J., Fernández-de Alba, J. M., and Fuentes-Fernández, R. Ambient intelligence with INGENIAS. In *Agent-Oriented Software Engineering XIII*, pages 118–133. Springer, 2013.
45. Gómez-Sanz, J. J., Fuentes, R., Pavón, J., and García-Magariño, I. INGENIAS development kit: a visual multi-agent system development environment. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Demo Proceedings*, pages 1675–1676. IFAAMAS, 2008.
46. Gómez-Sanz, J. J. and Fuentes-Fernández, R. Revisiting the Delphi method for agents. In *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Sustainability - The PAAMS Collection - International Workshops of PAAMS 2015, Salamanca, Spain, June 3-4, 2015. Proceedings*, pages 367–376, 2015.
47. Gómez-Sanz, J. J., Garcia-Rodriguez, S., Cuartero-Soler, N., and Hernandez-Callejo, L. Reviewing microgrids from a multi-agent systems perspective. *Energies*, 7(5):3355–3382, 2014.
48. Gómez-Sanz, J. J. and Pavón, J. Implementing multi-agent systems organizations with INGENIAS. In Bordini, R. H., Dastani, M., Dix, J., and Fallah-Seghrouchni, A. E., editors, *Programming Multi-Agent Systems, Third International Workshop, ProMAS 2005, Utrecht, The Netherlands, July 26, 2005, Revised and Invited Papers*, volume 3862 of *Lecture Notes in Computer Science*, pages 236–251. Springer, 2005.
49. Gómez-Sanz, J. J., Pavón, J., and Díaz-Carrasco, Á. The PSI3 agent recommender system. In Lovelle, J. M. C., Rodríguez, B. M. G., Aguilar, L. J., Gayo, J. E. L., and del Puerto Paule Ruíz, M., editors, *Web Engineering, International Conference, ICWE 2003, Oviedo, Spain, July 14-18, 2003, Proceedings*, volume 2722 of *Lecture Notes in Computer Science*, pages 30–39. Springer, 2003.
50. Hernández, L., Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., Cook, D. J., Chinarro, D., and Gómez, J. A study of the relationship between weather variables and electric power demand inside a smart grid/smart world framework. *Sensors*, 12(9):11571–11591, 2012.
51. Hernandez, L., Baladron, C., Aguiar, J. M., Carro, B., Sanchez-Esguevillas, A. J., Lloret, J., Chinarro, D., Gomez-Sanz, J. J., and Cook, D. A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. *Communications Magazine, IEEE*, 51(1):106–113, 2013.
52. Hiel, M., Aldewereld, H., and Dignum, F. Modeling warehouse logistics using agent organizations. In Guttmann, C., Dignum, F., and Georgeff, M., editors, *Collaborative Agents - Research and Development*, volume 6066 of *Lecture Notes in Computer Science*, pages 14–30. Springer Berlin Heidelberg, 2011.
53. Jensen, A. S., Spurkeland, J. S., and Villadsen, J. Formalizing theatrical performances using multi-agent organizations. In Jaeger, M., Nielsen, T. D., and Viappiani, P., editors, *Twelfth*

Scandinavian Conference on Artificial Intelligence, SCAI 2013, Aalborg, Denmark, November 20-22, 2013, volume 257 of *Frontiers in Artificial Intelligence and Applications*, pages 135–144. IOS Press, 2013.

54. Jiang, J., Aldewereld, H., Dignum, V., and Tan, Y. Compliance checking of organizational interactions. *ACM Trans. Management Inf. Syst.*, 5(4):23:1–23:24, 2015.
55. Jiang, J., Dignum, V., and Tan, Y. An agent-based inter-organizational collaboration framework: Opera+. In Cranefield, S., van Riemsdijk, M. B., Vázquez-Salceda, J., and Noriega, P., editors, *Coordination, Organizations, Institutions, and Norms in Agent System VII, COIN 2011 International Workshops, COIN@AAMAS 2011, Taipei, Taiwan, May 3, 2011, COIN@WI-IAT 2011, Lyon, France, August 22, 2011, Revised Selected Papers*, Lecture Notes in Computer Science, pages 58–74. Springer, 2011.
56. King, T. C., Li, T., Vos, M. D., Dignum, V., Jonker, C. M., Padget, J., and van Riemsdijk, M. B. A framework for institutions governing institutions. In Weiss, G., Yolum, P., Bordini, R. H., and Elkind, E., editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 473–481. ACM, 2015.
57. Koster, A., Madrenas, J., Osman, N., Schorlemmer, M., Sabater-Mir, J., Sierra, C., de Jonge, D., Fabregues, A., Puyol-Gruart, J., and García, P. u-help: supporting helpful communities with information technology. In *Proceedings of the First International Conference on Agreement Technologies (AT 2012)*, volume 918, pages 378–392, Dubrovnik, Croatia, 15/10/2012 2012.
58. Lee, J. *Norm Awareness for Virtual Characters Behaviour: A Socio-Cognitive Approach*. PhD thesis, University of Bath, January 2015. http://opus.bath.ac.uk/47059/, retrieved 2015-10-07.
59. Lee, J., Li, T., and Padget, J. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013.
60. Lee, J., Padget, J., Logan, B., Dybalova, D., and Alechina, N. N-Jason: Run-time norm compliance in AgentSpeak(L). In Dalpiaz, F., Dix, J., and van Riemsdijk, M. B., editors, *Engineering Multi-Agent Systems - Second International Workshop, EMAS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers*, volume 8758 of *Lecture Notes in Computer Science*, pages 367–387. Springer, 2014.
61. Li, T. *Normative Conflict Detection and Resolution in Cooperating Institutions*. PhD thesis, University of Bath, July 2014. http://opus.bath.ac.uk/45254/, retrieved 2015-10-07.
62. Li, T., Balke, T., De Vos, M., Padget, J. A., and Satoh, K. A model-based approach to the automatic revision of secondary legislation. In Francesconi, E. and Verheij, B., editors, *International Conference on Artificial Intelligence and Law*, pages 202–206. ACM, 2013.
63. Li, T., Balke, T., Vos, M. D., Padget, J., and Satoh, K. Legal conflict detection in interacting legal systems. In Ashley, K. D., editor, *JURIX*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 107–116. IOS Press, 2013.
64. Lopes Cardoso, H., Urbano, J., Rocha, A. P., Castro, A. J. M., and Oliveira, E. ANTE – A Framework integrating Negotiation, Norms and Trust. In *This volume*. 2016.
65. Lopes Cardoso, H., Urbano, J., Rocha, A. P., Castro, A. J., and Oliveira, E. ANTE: Agreement negotiation in normative and trust-enabled environments. In Ossowski, S., editor, *Agreement Technologies*, volume 8 of *Law, Governance and Technology Series*, pages 549–564. Springer Netherlands, 2013.
66. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.
67. Mensonides, M., Huisman, B., and Dignum, V. Towards agent-based scenario development for strategic decision support. In *Agent-oriented information systems IV*, pages 53–72. Springer, 2008.

68. Montero, R., de la Cruz, D., and Noriega, P. Prototipo de una plataforma de negociación on-line para el mercado de residuos. tr-iiia-2013-02. Technical report, IIIA - CSIC, Barcelona, 06/2013 2013.
69. Noriega, P. and de Jonge, D. Electronic Institutions. The EI/EIDE Framework. In *This volume*. 2016.
70. North, D. C. *Institutions, institutional change and economic performance*. Cambridge university press, 1994.
71. Padget, J., Elakehal, E. E., Li, T., and Vos, M. D. Inst*AL*: An Institutional Action Language. In *This volume*. 2016.
72. Padget, J., Elakehal, E. E., Satoh, K., and Ishikawa, F. On requirements representation and reasoning using answer set programming. In Bencomo, N., Cleland-Huang, J., Guo, J., and Harrison, R., editors, *IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2014, 26 August, 2014, Karlskrona, Sweden*, pages 35–42. IEEE, 2014.
73. Pavón, J., Gómez-Sanz, J., Fernández-Caballero, A., and Valencia-Jiménez, J. J. Development of intelligent multisensor surveillance systems with agents. *Robotics and Autonomous Systems*, 55(12):892–903, 2007.
74. Pavón, J., Sansores, C., and Gómez-Sanz, J. J. Agent based modeling of social systems. In *EUMAS*, 2006.
75. Pavón, J., Sansores, C., and Gómez-Sanz, J. J. Modelling and simulation of social systems with INGENIAS. *International Journal of Agent-Oriented Software Engineering*, 2(2):196–221, 2008.
76. Persson, C., Picard, G., Ramparany, F., and Boissier, O. A JaCaMo-based governance of machine-to-machine systems. In Demazeau, Y., Müller, J. P., Rodríguez, J. M. C., and Pérez, J. B., editors, *Advances on Practical Applications of Agents and Multiagent Systems, Proc. of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*, volume 155 of *Advances in Soft Computing Series*, pages 161–168. Springer, 2012.
77. Picard, G., Persson, C., Boissier, O., and Ramparany, F. Multi-agent self-organization and reorganization to adapt M2M infrastructures. In *Ninth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'15)*. IEEE Computer Society, 2015.
78. Pieters, W., Padget, J., Dechesne, F., Dignum, V., and Aldewereld, H. Effectiveness of qualitative and quantitative security obligations. *Journal of Information Security and Applications*, 22:3–16, 2015.
79. Pitt, J., Kamara, L., Sergot, M. J., and Artikis, A. Voting in multi-agent systems. *Comput. J.*, 49(2):156–170, 2006.
80. Quillinan, T. B., Brazier, F., Aldewereld, H., Dignum, F., Dignum, V., Penserini, L., and Wijngaards, N. Developing agent-based organizational models for crisis management. In *Proc. of the 8th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2009)*, pages 45–51, 2009.
81. Robles, A., Noriega, P., and Cantú, F. An agent oriented hotel information system. In Decker, K. S., Sichman, J. S., Sierra, C., and Castelfranchi, C., editors, *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)*, pages 1415–1416, Budapest, Hungary, 10/05/09 2009. International Foundation for Autonomous Agents and Multiagent Systems, ACM Press.
82. Robles, A., Noriega, P., Luck, M., Cantú, F., and Rodriguez, F. Multi agent approach for the representation and execution of medical protocols. In Moreno, A., Cortés, U., Annicchiarico, R., and Nealon, J., editors, *4th Workshop on Agents Applied in Health Care. ECAI 2006*, pages 11–16, 2006.

83. Rodríguez, J.-A., Noriega, P., Sierra, C., and Padget, J. FM96.5 A Java-based Electronic Auction House. In *Proceedings of 2nd Conference on Practical Applications of Intelligent Agents and MultiAgent Technology (PAAM'97)*, pages 207–224, London, UK, April 1997. ISBN 0-9525554-6-8.
84. Roloff, M., Stemmer, M., Hübner, J. F., Schmitt, R., Pfeifer, T., and Hüttermann, G. A multi-agent system for the production control of printed circuit boards using JaCaMo and Prometheus AEOlus. In *Proceedings of 12th IEEE International Conference on Industrial Informatics (INDIN 2014)*, pages 236 – 241. IEEE, 2014.
85. Scott, W. *Institutions and Organizations*. Sage Publications, 1995.
86. Serrano, E., Gómez-Sanz, J. J., Botía, J. A., and Pavón, J. Intelligent data analysis applied to debug complex software systems. *Neurocomputing*, 72(13):2785–2795, 2009.
87. Sorici, A., Picard, G., Boissier, O., Santi, A., and Hübner, J. Multi-agent oriented reorganisation within the JaCaMo infrastructure. In *The 3rd International Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS 2012)*, 2012.
88. Sánchez-Anguix, V., Esparcia, S., Argente, E., García-Fornes, A., and Julian, V. Collaborative information extraction for adaptive recommendations in a multiagent tourism recommender system. In *8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010)*, volume 70, pages 35–40. Springer, 2010.
89. Thompson, M., Padget, J., and Battle, S. Governing Narrative Events With Institutional Norms. In Finlayson, M. A., Miller, B., Lieto, A., and Ronfard, R., editors, *6th Workshop on Computational Models of Narrative (CMN 2015)*, volume 45 of *OpenAccess Series in Informatics (OASIcs)*, pages 142–151, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
90. Toledo, C. M., Bordini, R. H., Chiotti, O., and Galli, M. R. Developing a knowledge management multi-agent system using jacamo. In *ProMAS*, pages 41–57, 2011.
91. Trescak, T., Esteva, M., and Rodriguez, I. A virtual world grammar for automatic generation of virtual worlds. *The Visual Computer*, 26(6):521–531, 2010.
92. Westra, J., Dignum, F., and Dignum, V. Keeping the trainee on track. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 450–457. IEEE, 2010.
93. Westra, J., van Hasselt, H., Dignum, F., and Dignum, V. Adaptive serious games using agent organizations. In *Agents for Games and Simulations*, pages 206–220. Springer, 2009.
94. Wooldridge, M. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
95. Yaich, R., Boissier, O., Picard, G., and Jaillon, P. Adaptiveness and social-compliance in trust management within virtual communities. *Web Intelligence and Agent Systems*, 11(4):315–338, 2013.