A Novel Approach to Modelling Distributed Systems - Using Large-Scale Multi-Agent Systems

Bogdan Okreša Đurić

Artificial Intelligence Lab, University of Zagreb
Faculty of Organization and Informatics, Pavlinska 2, 42000 Varazdin, Croatia

Abstract Large-scale multi-agent systems (LSMAS), a rather novel concept in the domain of multi-agent systems (MAS), are reaching for soaring heights in the wake of the Internet of Everything era. Interacting pieces of software on interconnected machines, enabled by rapid development of the Internet and connected devices, are creating systems comprising tens of thousands, even millions, of agents. Each agent is thus situated in an environment with numerous other elements and interaction is inevitable. Such a situation benefits from organisational modelling of the system. Using an organisational metamodel, which provides concepts for definitions of several organisational models, introduces upgraded time and effort efficiency into LSMAS organisational modelling, thus aiding in cost and time efficiency of design and development of distributed software. This chapter introduces a novel method of LSMAS organisational modelling using an organisational metamodel which makes it easier to model an LSMAS at various levels of abstraction. The presented metamodel is a work-in-progress description based on an ontology being developed that comprises LSMAS organisational concepts. Some features of the metamodel are presented, in this chapter, using two distinct examples of LSMAS application domains. Main features differentiating the proposed metamodel from the existent LSMAS organisational models also include concepts for modelling interorganisational dynamics.

1 Why it is Important to Consider Organisation

Recent developments introduced by the modern technologically advanced era have led to increased use of virtual (software) agents as opposed to using real-life agents i.e. people in various scenarios. People have always been attracted to the idea of organisation. Groups of people were formed ever since *homo sapiens*, and our ancestors, discovered benefits of socialisation, either in a planned manner or motivated by a common need of some kind, e.g. shelter, defence, hunger. Such organisations consisted of anything from only a couple of individuals (e.g. ancestors of the modern *homo sapiens* in search of fulfilment of the mentioned needs) to as many as needed for a great army of the 20th century.

Organisation applicable to the mentioned examples can be perceived as having one main function — overcoming various limitations of individual agents. [48, 62, 63] These limitations have several aspects, e.g. temporal (one agent has temporally limited availability), functional (some actions demand simultaneous effort, and an individual agent may not be capable of offering such a functionality), etc. Speaking of organisation among agents in a multi-agent system (MAS) implies a MAS comprising intelligent agents that can: interact with each other and the environment, reason, act and react upon their perceived environment, communicate with each other, observe changes in the system, etc. It has been a prevalent thought in MAS-related studies that such agents can successfully serve as models of real life situations and real-life people. Likewise, it is considered that human organisations, and principles of human organisation, can be successfully modelled using interactive intelligent agents. Although modelling agents conforming to their real-life counterparts can be a tough goal to achieve, it is rewarding in the long run, since various experiments can be conducted within lower budget capacity, shorter time periods, and with greatly increased reproducibility.

Organisation is not the only way to structurally model a system of agents. Another popular way of structured interaction and functioning of a MAS is swarm intelligence. While organisation model is derived directly from the well-known and researched concept of human organisation, swarm implies close ties to swarms of insects or similar forms of life. Each of these approaches to building MAS (or even large-scale MAS (LSMAS)) presents their researchers with different features of the resulting system, and is thus more suitable for a specific application. Organisational modelling on one hand is about defined structure in the system followed by structured communication protocols in terms of hierarchy or possible ways of communication flows. Furthermore, organisational ideas can be known by individual agents directly, or they can be imposed on the given system, with agents being aware of the enforced elements of organisation, or the included agents can be ignorant of the system's organisational constraints, etc. Swarm [18, 41], on the other hand, is all about self-organisation and emerging organisation. Imagine a swarm of bees [8], a school of fish [40], or a colony of ants — there is no specific entity that would govern their behaviour, rather they act as part of a group which benefits from their individual behaviour. Structure therefore emerges from the group, based on performance of individual agents. The main difference, when observing organisation, is in the way organisation elements are formed — starting at the level of an individual agent (bottom-up), or from the level of the whole given system (top-down).

Each of these approaches is beneficial to specific scenarios, depending on many variables, including the type of environment of the modelled agents, the type of agents and their abilities, the main goal of the system, etc. Therefore, an extremely biased approach may not be the best way to model a system. [13, 43] Furthermore, it is rather easy imagining an example where both approaches intertwine, e.g. interaction of the swarm creates a certain organisation-like structure which is propagated further and strengthened until the swarm, probably provoked by another need of theirs, in a non-unison way, decides that organisational dynamics is in order, and the structure changes, if only for a small amount.

Backed by a notable development of computer power in the past few decades, rising popularity of agent-based structures and agent-aided distributed computing can be attributed to the rising complexity of software problems and e.g. use of computing for conducting research on a global scale. From a different perspective, agent-based distributed computing is very beneficial to, and benefits from, the rising number of individual computer-imbued things a person can possess. Smart cars, smart bicycles, smart phones, smart cups, smart homes, smart cities — the potential for connecting all the existing pieces of software that are capable of connecting to e.g. the Internet, is obvious. It may be seen as most advantageous to observe these pieces of software residing in many household things as agents, and the Internet, or a local network comprising these agents, as a MAS. Furthermore, it is argued that the efficiency of such systems is raised using structured organisation, imposed upon the system, since it benefits more from the existent number of agents, their possibilities, and their focus on achieving a joint goal. [22, 25, 29]

The mentioned scenario forms a basis for the Internet of Things (IoT), or in an even more general case, and of larger scale, the Internet of Everything (IoE). While IoT is clearly concerned with things and objects that are able to interact and cooperate with each other to reach their common goals [48], thus creating a rather clear possibility of being abstracted as MAS, IoE covers a much wider domain comprising people, processes, data and things working together to make appropriate and beneficial connections; more so than ever before. [48]

Examples of IoE or IoT paradigms are applicable to various domains. Recent studies at the Artificial Intelligence Laboratory of Faculty of Organization and Informatics at the University of Zagreb (AI Lab @ FOI) studied smart cities, and agents in a massively multiplayer online role-playing game (MMORPG). When thinking in terms of smart cities or MMORPGs as application domains of LSMAS, it is favourable to think about organisation of included agents. Furthermore, it is advantageous to import various features of a human organisation into a system of agents.

Smart planning is a crucial element in planning and realisation of a project. There are several modelling methods, the most popular of which may be the UML notation, but only few are designed especially for MAS, let alone LSMAS. Organisational modelling of an LSMAS may be considered as planning a system of agents.

An organisational metamodel for modelling of LSMAS is being developed at the mentioned AI Lab. The general idea and goal of this research is to develop an extensive model that would encompass several different organisational models and structures (e.g. horizontal vs. vertical organisation). Such a model will utilize a clear visual representation of the modelled concepts, and will make it easier to plan an LSMAS, since it will incorporate elements of various perspectives of organisational modelling, e.g. organisational culture, strategy, or organisational change. Furthermore, it will be possible, when the model is finished, for the user to generate a programming code skeleton, based on the built model.

The approach just described will make it possible for the user to build a model where most of the elements of the future system are specified. This step of creating an LSMAS is of great significance, wherefore the approach that favours change and alteration is most welcome, and this metamodel will offer one such approach. Visual design of a model will surely make it easier for the user to review the built model, and to incorporate changes identified when evaluating the model built. Usefulness of code-generating part of the metamodel is obvious with respect to definitions built in the model. Since the code-generating process is automated, and based on the built model, the outcome is bound to be represented by the said model built by the user. The metamodel is envisioned as a rather abstract view of the system though. Therefore, the generated computer code will represent only basics, and the programmer is expected to fill in all the needed detail.

The role of an organisational (meta)model is therefore obvious in planning and development stages of a distributed computing software project that relies on agents and their interaction.

The rest of the chapter is represented as follows. The rest of this section contains further details about MAS, emphasizing roles of IoE and MMORPG and how they are related to LSMAS. Some basic organisational elements observable, and beneficial to, IoE and MMORPG will be noted as well. A brief overview of organisation of MAS and, more specifically, LSMAS, will be given in Sect. 2. Section 3 covers two distinct use cases for the proposed organisational metamodel, repercussions of which, along with feedback, are discussed in Sect. 4. Brief overview and guidelines for further research are covered in Sect. 5.

1.1 About the Internet of Everything and Massively Multiplayer Online Games

It was mentioned earlier that, in their most basic form, an agent is a software entity surrounded by an environment. An interactive intelligent agent can perceive this environment of theirs, and act upon it. [45] Such an agent can be considered a virtual representation of a human in a group or a system.

Although multi-agent systems represent an area where a lot of research has been done already, their larger counterpart, LSMAS, has had some research done only recently. Probably the most well-known concepts where LSMAS may be applied are the application areas of IoT [4, 55] and IoE. Even though IoT and IoE are used almost synonymously, there is a slight difference between the two, as nicely put by Cisco [36]: *In terms of phases or eras, Cisco believes that many organizations are currently experiencing the Internet of Things (IoT), the networked connection of physical objects. As things add capabilities like context awareness, increased processing power, and energy independence, and as more people and new types of information are connected, IoT becomes an Internet of Everything — a network of networks where billions or even trillions of connections create unprecedented opportunities as well as new risks*.

IoE seems to be the inevitable future of distributed computing, and the core idea of distributed systems. Furthermore, some indications of a novel concept of the Internet of Agents appeared recently, e.g. [61]. A notable IoE area of application are smart cities. [51–53, 57] When thinking about a city filled with agent-controlled elements (e.g. cars, traffic lights, parking lots, buildings and homes, etc.), it may do well to think about organisation features amongst all the included agents. As opposed to swarming agents and emerging organisation traits based on behaviour of agents, agents in a city would be demanded from and *per se* inclined to fulfil a given task in an optimal amount of time, using the least resources and in the safest way possible. Such a task undertaken by every of thousands of agents would greatly benefit from features mirrored from human organisations, such as communication protocols, rules of conduct, and similar. IoE, abstracted by LSMAS in a way similar to the one described may be applied to other domains, e.g. smart power grids, smart health, smart transport, smart buildings, etc. where some of the elements may be considered as sub-elements of e.g. smart cities.

IoE is a rapidly developing area that can be abstracted by LSMAS. Another example of great interest is domain of massively multiplayer online games (MMOGs). An MMOG is a computer game meant for a great number of players simultaneously playing the game online, often engaged in interaction with each other. MMORPG is a special kind of an MMOG that allows players to take control of their avatar (in-game character of the player) and interact with usually vast virtual worlds where many automated (non-player) characters and other players' characters reside. [49] Such games represent proper LSMAS environments: there are numerous agents (some controlled by players, most acting independently) with many opportunities to interact (e.g. trading, combat, training, pillaging, cooperation, communication, delegation, etc.) and a big world to explore (sometimes consisting only of towns, areas and cities, but some expand to planets and solar systems). In order to succeed, players often have to cooperate, i.e. join in smaller or larger groups, where they have to exercise real-life-like interaction with others, including choosing a leader, following orders, or planning an attack. Such worlds are obviously very interesting grounds for training and experimenting with agents in an LSMAS.

2 Overview of Models for Organising Agents and State of the Art

As mentioned by several studies, only some of which are [10, 25, 29, 31], imposing organisation features on an LSMAS may bring more benefit to the system, than using the agent-centred paradigm. Therefore, it is interesting to talk about organising systems of agents. As mentioned earlier, copying elements of human organisations, and applying them to artificial agents is the prevalent method of developing organisational features for systems of agents.

In this section an overview of organisational aspects meant for systems of agents is followed by a modern view of organisational modelling, needed for modern systems of large scale, as proposed in recent studies on LSMAS and the IoE.

2.1 Existing Models for Organisation of MAS

Organisational models and frameworks for organisational modelling of MAS have usually concentrated on structural features of an organisation. Organisational structure, as a primary feature being modelled, is often accompanied by concepts used for modelling functional aspects of an organisation, and concepts which aid in modelling agent interaction within a MAS. Other organisational features, such as norms, or the environment in which agents are situated, are rather scarce in the MAS organisational models and frameworks developed to date.

This section covers an overview of some of the more popular means of modelling organisation of MAS, as shown in Table 1, along with their most significant dimensions (features) as described by Coutinho et al. in [14].

Table 1 Models and frameworks for organising MAS and their respective dimensions, according to the containing concepts, as described in [14].

Organisational model	Dimensions
AGR	Structure, interaction, agents
TÆMS	Functions, processes, environment

MOISE+	Structure, functions, norms
ISLANDER	Structure, norms, interaction
OperA	Structure, functions, norms, interaction
AUML	Structure, functions, interaction, environment
NOSHAPE MAS	Structure, dynamics, agents
MACODO	Structure, dynamics, agents

Models mentioned in Table 1 are further described by their authors in their respective papers. Some basic pieces of information about concepts used by each of the stated models are laid out further in this chapter. Such an overview is an introduction to perspectives of LSMAS organisational modelling.

AGR The *Agent/Group/Role* model, or AGR model in short, was developed by Ferber et al. [22], and is also known as *Aalaadin* model. The three basic concepts included are meant for modelling individual agents, groups of agents and agent roles. The concept of agent within AGR conforms to features of agents mentioned earlier – it denotes individuals capable of interacting and communicating that can range within both extremes of reactivity and intelligence. The main trait of these agents, as the model is not concerned with their internal architecture, is that an agent plays roles and belongs to groups. A group consists of many agents that share a common interest or a characteristic. Thus a group can be used for creating organisational segments, and functional or structural parts of an organisation.

TÆMS As a framework developed by Decker, presented in [16], originally intended for modelling of complex computational tasks, *Task Analysis, Environment Modeling, and Simulation* framework can be used with MAS as well. The most prominent feature of TÆMS related to MAS is layered description of environments (not exclusively of the same meaning as environment in MAS). Closely connected to concepts describing environment are concepts for statements about tasks and task groups. Three layers described in [16] (objective, subjective, and generative) are defined as levels of environmental and task characteristics model. It is interesting to note that TÆMS models an agent as a locus of belief and action.

MOISE+ Building on *Model of Organization for multI-agent SystEms* (MOISE), and *Aalaadin*, both organisation centred models, MOISE+ comprises concepts for structural, functional, and deontic specification of organisation in a MAS. Although direct modelling of agents is not possible, MOISE+ depends on modelling roles, relations amongst them, and groups. [30] Roles represent constraints individual agents must follow when playing a specific role. Possible roles an agent can play depend on the roles the given agent is playing already. Upon accepting to play a role, the given individual agent is added to a group playing that specific role. Another point of interest in MOISE+ is functional specification, wherein goals are structured in plans, and grouped in missions. It should be mentioned that notation for sequential, parallel, and choice-based plans is present.

ISLANDER Seemingly situated slightly off of the centerpoint of MAS modelling, ISLANDER is a language for textual specification of electronic institutions. [20] Main parts of the language are used for specifying performative structure, scenes that make up the said structure, and normative rules. Scenes serve as meeting points for agents communicating according to well-defined protocols. Roles again represent specific constraints over individual agents, along with specifying their possible actions (e.g. communication protocol). Normative rules set up agent actions that have consequences of some gravity.

OperA This framework developed by Dignum presented in [17] is primarily focused on describing system at a conceptual level. Therefore, the developed concepts are mainly used to define structure and global behaviour of the model, including e.g. organisational characteristics, while individual agents that populate the said model are modelled separately and independently of their internal design. Such a feature is achieved using three components: organisational model, social model, and interaction model. The organisational model encompasses concepts of roles and interactions, the social model populates the defined organisational structure with agents playing roles, and the interaction model is built using interaction between agents.

AUML During the year 2001 an effort was made, described by Van Dyke Parunak and Odell in [19], in order to enrich *Unified Modelling Language* (UML) with concepts useful for agent-based systems (i.e. MAS). Concepts that were identified as most useful are: swimlanes, class diagram, sequence diagram, and activity graph. Swimlanes were proposed as representation of groups of roles, along with role instantiation. Class diagrams were used to define roles and their relationships, similar to swimlanes enhanced by cardinality constraints. Sequence diagrams were used to describe possible interaction amongst various agent roles. In the end, interaction of groups and group-level dependencies, where these groups can be modelled as agents, was shown using an activity graph.

NOSHAPE MAS The main purpose of this novel organisational model is to be the most general one of those mentioned here. NOSHAPE recognises three levels of abstraction: universe, world, and organisation. Using concepts of holarchy and hierarchy, Abbas [1, 2] thinks of agents as individuals or groups depending on the perspective: bottom-up perspective sees a group, while top-down perspective is concerned with agents as individuals. Therefore, levels of abstraction consist of several individuals of lower level abstractions (e.g. a universe comprises an infinite number of worlds). Each of these agents are situated in an environment, and can interact with each other. Naturally, the concept of roles is existent as well. An interesting observation are static roles, such as Global Supervisor and Local Supervisors - roles that are concerned with organisational structure or organisational behaviour.

MACODO This organisational model used for describing dynamic organisations is a part of an integrated approach called *Middleware Architecture for Context-driven Dynamic agent Organizations* (MACODO). [58, 59] The main feature of this model is that agents are modelled separated from their life-cycle, thus making it easier to understand, and model, how changes in the system, or changes in the environment, affect dynamic organisations, i.e. agents. Agents are uniquely identified within the system, and have their capabilities grouped into sets called roles.

This brief overview of some of the better known organisational models or framework indicates that said models, as shown in Table 1, usually comprise concepts describing organisational structure of a system of agents (e.g. groups of agents), interaction of agents (e.g. communication protocols), normative restrictions (e.g. norms in context of constraints over agents and their capabilities or rules of conduct), functional features of an organisation (e.g. capabilities of agents), etc. All the mentioned models, except the most recent one, NOSHAPE MAS, are concerned with MAS in general, without mentioning LSMAS in particular. Only NOSHAPE MAS mentions several levels of abstraction, and thus potential for a large scale organisation.

2.2 Recent Advancements in LSMAS Organisational Models

As was mentioned earlier in this chapter, multi-agent systems of large scale have been recently shown as applied to the Internet of Things, or the Internet of Everything. One such example is coming from the medical area, specifically distributed worldwide health care applications, as described by Bui and Zorzi in [11]. The mentioned authors strive to create a communication framework for agents included in the system. This permits argument about communication methods of agents within LSMAS, and requirements of that particular element of organisation.

When speaking of distributed systems consisting of autonomous agents, Scheutz noted in his 2010 research [50] that MAS did not have enough flexibility at the time, nor were they supportive enough, for sophisticated large-scale intelligence applications. The solution the mentioned author proposed was in synergy of MAS and system of single agents.

A rather long time ago in context of LSMAS, in 2002, McCauley and Franklin described an LSMAS used in US navy personnel distribution [35]. The described system looked after the needs of US navy entities taking care of e.g. their needs, state of the system, scheduled personnel changes, etc. Three main classes of agents existed: sailor agents, command agents, and navy agents. Most of the communication and interaction took place between sailor and command agents (they negotiate available positions, sailor interests, etc.), while the navy agent acts as an overseer.

On the other side, and published more recently, research was done by Schatten that takes into account the interdisciplinary potential of LSMAS research. [46] The mentioned author uses complex analytical method (cro. kompleksna analitička metoda, KAM) to conduct self-organisation in MAS. In general, KAM is used to analyse organisations and propose organisational model that is new and optimised. Using KAM, and adapting it to MAS, is enhanced using the fractal principle (e.g. every agent is considered an organisational unit, but a group of agents that collaborate and have a common goal, are considered an organisational unit as well). Such an approach is rather similar to ideas of holons and holarchy [3].

Another proposal intended to create an easier way to work with agents in LSMAS is described by Boulaire et al. in [9]. The mentioned authors propose an approach of dynamic agent composition, that is intended to break agents into atomic units (i.e. parts) that together form a complete agent, and the whole system, and are combined at runtime. The three entities that form an agent are an asset, behaviours, and data. This novel approach to building agent-based models (ABMs) aims to extend ABMs with underlying networked structure, thus allowing users to develop new elements of a system, and add them to an existing system, without the need to access or modify previously written code.

Further research done by Schatten as elaborated in [47] is even more pertinent to large-scale of MAS, and foundations are set for an ontology comprising concepts of organisational modelling applicable to the domain of LSMAS. This approach, of creating an initial ontology for modelling complex systems, is deemed necessary by the mentioned author, since it would allow for definition of formal semantics of the modelled systems. Furthermore, some conceptual foundations are laid for an LSMAS framework.

This research done by Schatten [47] provides several perspectives of organisational modelling that are recognised as crucial in future development of LSMAS, i.e. organisational models of LSMAS. Some of the concepts of such an ontology of organisational design methods are detailed in the chapter as well. The said perspectives are defined by Schatten [47] as follows:

- Organizational structure defines the decision and information flows of an organization.
- Organizational culture defines important intangible aspects of an organization including knowledge, norms, reward systems, language and similar.
- Strategy defines the long term objectives of an organization, action plans for their realization as well as tools on how to measure success.
- Processes define the activities and procedures of an organization.
- Individual agents define the most important asset of any organization - the individuals actually performing the work.
- Organizational dynamics define organizational changes including reorganization of any of the above mentioned components.
- Context and inter-organizational aspects define organizational behaviour towards its environment including strategic alliances, joint ventures, mergers, splits, spinouts and similar.

Some of the most recent studies [5–7, 15, 21, 24, 26–28, 32, 34, 36, 42, 44, 54, 56, 60] provide further incentive to think of research on systems of agents, especially those of large-scale, as important.

A clear direction of thought is recognisable in some of the models mentioned in Sect. 2, as the following can be observed:

- many of the mentioned models think of MAS and LSMAS on a number of levels of abstraction (since LSMAS may comprise thousands of individual agents, it may seem like a natural way of viewing such systems);
- somewhat of a leitmotif is use of roles for introducing constraints or a set of features for individual agents;
- grouping agents by roles is used often;
- it is curious that only the most recent studies think of dynamically changing systems.

3 The Metamodel and Examples of its use

The organisational metamodel that is used in the following examples is being developed [37, 49] based on recent studies mostly presented in this chapter. The finished metamodel will be based on an ontology being developed by the author of this chapter. [38, 39] An overview of the metamodel at this very early stage of development is given in this section, followed by two examples of its use.

The following examples shall be used to demonstrate the early working version of the organisational metamodel for LSMAS being developed. The main idea of this metamodel is to adhere to the seven perspectives of organisational modelling of LSMAS mentioned by Schatten in [47], that are described informally in Sect. 2.2 of this chapter. Therefore, the individual agents are regarded as organisational units. Furthermore, an organisational unit can comprise infinite number of organisational units. What is even more interesting, such a relation will be used for even more organisational entities of the system being modelled, e.g. goals, tasks, etc. The metamodel in its final version is expected to include, amongst others, concepts related to organisational change, and, the most complex element, organisational culture.

At the moment, the metamodel can be used for modelling the following concepts of an LSMAS: Organisational Unit, Role, Goal/Objective, Process, Knowledge Artefact, and several properties of these concepts, e.g. inclusion, possible Roles of an Organisational Unit, command flow, etc. These organisational concepts have been identified upon analysis of an ontology containing selected organisational concepts used in organisational modelling of LSMAS. [38, 39] The ontology is a work-in-progress as well, and is being built based on standard practices of organisational modelling of human organisations, but is clearly directed towards LSMAS. The aim of the finished metamodel is to comprise LSMAS organisational concepts identified based on the mentioned ontology and on the related research, all of which shall be in accordance with the modern features of LSMAS organisational modelling [47]. Distinction of the proposed metamodel, when compared with existent LSMAS organisational models, will be visible through elements such as included concepts for modelling interorganisational dynamics, and a “zoomable” approach, where many of the included concepts will be observable on various levels of abstraction, to name a few.

An Organisational Unit (see Fig. 1a) is the basic element which represents an individual agent. Alternately, an Organisational Unit can represent a type of agent (see difference between approaches in Sect. 3.1 and Sect. 3.2 below). Every Organisational Unit can access an Individual Knowledge Artefact (Fig. 1b), and it can play any number of Roles. Furthermore, an Organisational Unit can be a part of another Organisational Unit (Fig. 1d), thus creating a group, or it can answer to another Organisational Unit (Fig. 1c). Organisational Units are not supposed to be detailed any further, since the emphasis of the model is on organisational, and not individual modelling.

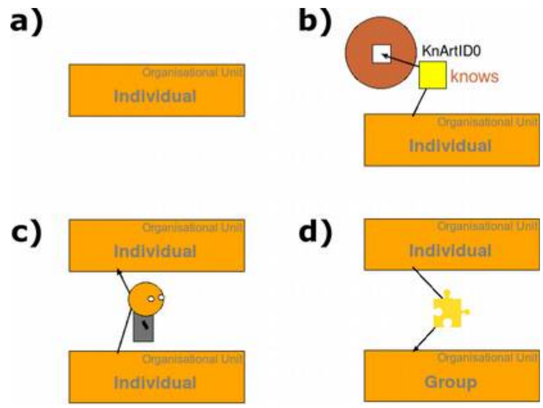

Fig. 1 Visual representation of the Organisational Unit concept

Roles are modelled with the idea of constraints in mind. Every Role (Fig. 2a) has some dedicated actions which become available to an agent that plays it. Furthermore, Roles have basic properties similar to those of Organisational Units: hierarchical command flow (Fig. 2d), grouping relation, access to Organisational Knowledge Artefacts (Fig. 2b). As opposed to an Organisational Unit, a Role can combine its available actions in a Process that can be used to achieve a certain Goal (Fig. 2f). Every Role can have a specific Goal (Fig. 2c) that can hierarchically consist of subgoals. Finally, a Role can have a generic relationship with another Role (Fig. 2e). Such a property allows for the developer to specify the connection they need. A Role is thus somewhat of a central concept in modelling a system. Certainly, the approach depends on the will of the developer, and purpose of the model.

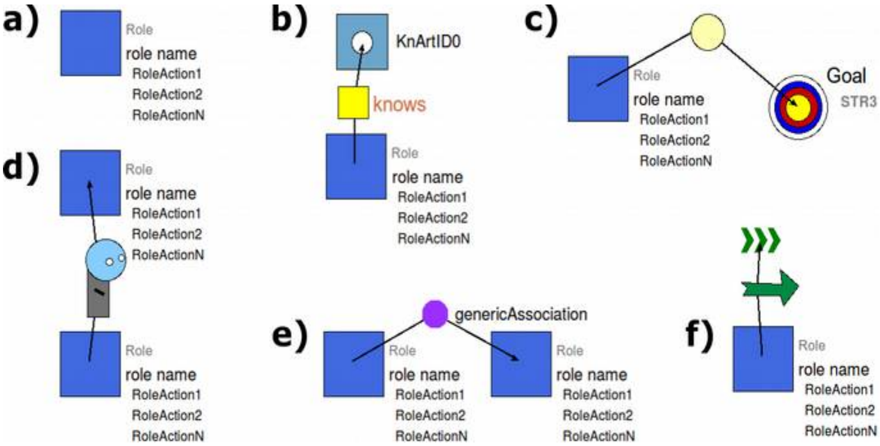

Fig. 2 Visual representation of the Role concept

A Knowledge Artefact contains a piece of knowledge of the system. Modelled as abstract representations of knowledge, Knowledge Artefacts are designed to be detailed by the developer once the system development starts after the modelling phase. Knowledge Artefacts (KnArt) are divided into Individual and Organisational KnArts. Individual KnArts (Fig. 1b) contain knowledge of importance to individual agents. Organisational KnArts (Fig. 2b), on the other hand, represent pieces of knowledge pertaining to the organisational aspects of the system, and are, by default, accessible to Roles only.

Goals, i.e. objectives, are modelled to contain specific basic information about the given goal, and what is needed for the goal to be fulfilled. Therefore, every Goal can have its respective Measurement and Reward values (Fig. 3). These are written in the way most apt for the development process of the system. Every Goal may be a part of another Goal concept, thus creating a hierarchy and subgoals.

A Process concept is abstracted as a concept that can be enacted by a Role, and that has a certain Goal concept for its outcome. Therefore, the outcome can be achieved using the designated Process. A Process can be used to achieve a certain subgoal as well, thus being useful in fulfilling complex Goal concepts. Ideally, a Process available to a specific Role consists of actions available to that particular Role.

Properties included in the metamodel will not be detailed here, as their basic information was provided above, when other concepts were described in more detail.

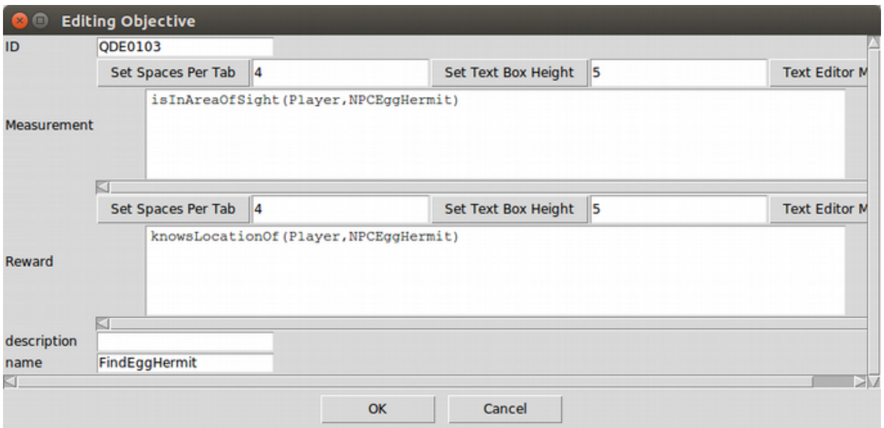

Fig. 3 Details of a Goal concept are described using predicate logic statements

3.1 A Simple Example of the RecipeWorld

The RecipeWorld was developed by Fontana and Terna as an "agent-based model that simulates the emergence of networks out of a decentralized autonomous interaction." [23] This model is built using three basic elements: recipes, orders, and agents. Recipes contain a custom number of steps that have to be taken if an objective is to be achieved. Orders are specific objectives that have to be fulfilled. Every order has some technical information, and auxiliary data. Finally, agents are solving given problems, by completing steps defined by a recipe. Technical details (e.g. recipe structure) can be found in [23]. As noted by the mentioned authors, one of the goals of this agent-based model is to generate network based on activity of agents, instead of making agents generate a network a priori.

Typical application example of the described model is that of production. Several factories have to produce all the generated orders, thus creating a social network that can be analysed for specific insight into the production process. Additional constraints are introduced into the system (e.g. a specific factory can only work on a specific element of a recipe). Each factory of this system, and each order to be produced are represented as agents. As the system is run, production is started, and the orders move around the system to factories that can produce the needed recipe part, and thus a network is generated.

In this example, individual structure, or functioning of an agent is not of great concern, i.e. an agent will be modelled almost as a black box, giving the system designer freedom to develop the agent in any way they want. Each agent will be given certain constraints though. First of all, roles will be used to determine if an agent is a factory, or an order. This way, a role will contain a set of constraints, and the agent playing the given role will have to act accordingly. Furthermore, knowledge of a recipe of the order will be modelled as a knowledge artefact. This way, the abstract concept of a knowledge artefact can be realised by the system developer in a way they desire (e.g. a rule language like RIF or SWRL). In order to utilize and simplify communication of agents involved in the system, a knowledge artefact specifying ontology of communication concepts will be accessible to all the roles of the system. Every role in the system will be related to a couple of needed functions or processes as well. This way, the whole process of organisational design will be moved to a more abstract layer, as opposed to working with individual agents. What is more, this type of declaration allows for computer code generation of the basic elements of the modelled system. The described process is detailed as follows.

One version of a situation modelled as described is show in Fig. 4. Individual agents are represented using orange rectangles. An individual agent can play any of the connected roles represented as blue squares. Furthermore, as described above, every agent has individual knowledge of their recipe parts, or services they provide, based on the designated role they will be performing. This formulation presumes that individual agents performing different roles will be basically different. Every role has several actions that are clearly named, and will most likely be translated into code. As mentioned above, both roles have access to a kind of a knowledge repository, a knowledge artefact of organisational concern, which stores the domain ontology. Each role has its respective main goal (i.e. objective) which is decomposed further. Subgoals are not defined as goals of the given role, but are achieved by processes available to a specific role. Two separate processes are available to the Factory role, and their result should be fulfilling the goals they are connected to.

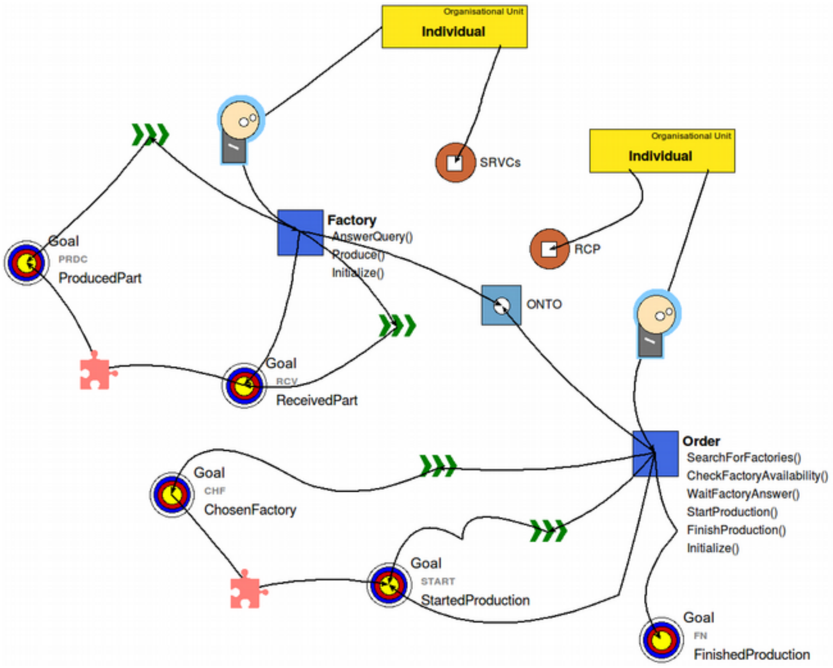

Fig. 4 Sample representation of the RecipeWorld by the organisational metamodel of this chapter

Model presented in Fig. 4 may yield programming code as follows. It is worth noting that the metamodel is a work in progress, and code generation is one of the features not yet implemented. Therefore, only a possible suggestion based on the model is shown. The programming code shown in the box below is based on Python and SPADE (Smart Python multi-Agent Development Environment), where agents are instances of an agent class, and their behaviours are instances of behaviour classes covering various types of behaviours.

```
 class AgentOrder(spade.Agent.Agent):
 class SearchForFactories(): […]
 class CheckFactoryAvailability(): […]
 class WaitForFactoryAnswer(): […]
     class StartProduction(): […]
     class FinishProduction(): […]
     def initialise ():

class AgentFactory(spade.Agent.Agent): […]
   class AnswerQuery(): […]
     class Produce(): […]
   def initialise (): […]
```

Further details from the model, e.g. knowledge, measurement and rewards of goals, are not shown in this example code, as they may be realized using various tools, most useful of which may be combination of RDF (Resource Description Framework) and OWL (Web Ontology Language), combined together into an ontology, since they are well adapted to the task of modelling knowledge.

3.2 A More Complex Example from an MMORPG

The second example that will be given in this chapter is devised to show how adaptable the proposed organisational metamodel is to the scale of the modelled system. The main difference between this example and the previous one is in scale, e.g. in the number of possible roles within the system, in the number of active individual agents, in the number and diversity of tasks and goals, in the possible combinations of all the included elements of the system.

MMORPGs, and MMOGs as a more general concept, are a good application domain for LSMAS as they can engage hundreds, thousands, and even millions of players. Many popular examples prove this (e.g. League of Legends, Hearthstone, Dota 2, World of Warcraft, etc.), and such games are still gaining popularity, presently having millions of regularly active players. [49] Furthermore, MMOGs are interesting to research [33], since they are interesting to players that are eager to explore and interact with a virtual world, simultaneously motivating them to communicate, and cooperate or fight with other players or elements of the environment. Group elements (including organisational features and social skills) are usually essential in games of MMORPG genre, since it is often impossible for a player's avatar (in-game character controlled by a human player) to survive or be successful in the given virtual world on their own.

The following example is based on an MMORPG scenario developed for the purposes of the Large-Scale Multi-Agent Modelling of Massively On-Line Role-Playing Games (ModelMMORPG) research project of the Artificial Intelligence Laboratory (AI Lab) of Faculty of Organization and Informatics at the University of Zagreb. The scenario was based around a developed quest named The Quest for the Dragon Egg. In order to accomplish this quest, a player had to retrieve a Dragon Egg item from one of the three dragon dens located throughout the virtual Mana World, but the exact location of the egg was a secret, as was the precise time when the egg was going to spawn (with the interval being about 24 hours). Upon finding the Dragon Egg item, after having fended off about a dozen Dragon monsters guarding it, a joint effort was necessary to transport the Dragon Egg item to the safe place. At least three player avatars (player characters) had to be present in each other's line of sight at any one moment while the Dragon Egg item was being transported, or the quest would fail. That was not the end though. In order to receive the main prize of the quest, and to actually solve it, the egg had to be hatched using another special item, a Hatching Potion. This potion had to be made using several specific ingredients, making it another group effort. Only upon bringing the Hatching Potion and the Dragon Egg item, within a specified period of time, to a specific non-player character (NPC), a friendly Dragon monster could be spawned, and the quest finished. This devised quest is a clear representation of how important cooperation is in MMOGs, especially MMORPGs. It is important to note that only leaders of a group of players (called a party) could initiate the quest. Once the quest was initiated, only members of the initiating party could participate in the quest, and gain benefits of the solved quest. The key element in analysing the mentioned quest, and modelling it, is the fact that inclusion of many individual agents does not change the amount of set constraints in form of a role or any other concept.

An avatar is an individual agent in the scope of this example, so it shall be represented as an organisational unit. Since one of the key elements in the quest is a group of players, another organisational unit shall represent a party. Notice slightly different way of modelling, when compared to the previous example, since the emphasis will be on roles. Indeed, it is more interesting and useful to develop roles for the scenario described in this example. Furthermore, Fig. 5 shows only a part of the whole system, i.e. the part that describes elements, and their relationship, which are the closest to a player's avatar.

Individual agents are able to play several specific roles, e.g. gatherer, fighter, support, herbalist, transporter, scout, DragonEggQuester, etc.

Every role has specific processes it can use in order to achieve specific goals. The main objective of the quest is designed as an objective of the DragonEggQuester role, and has a specified reward obtainable if finished. As such, the objective is divided into several lower-level specific quests. An individual agent is expected to use a role, and processes found therein, that is most suitable for reaching the identified specific goal.

State of the world, and of individual agents, is defined using knowledge artefacts. Specifics about various concepts found within the game are detailed in a connected ontology.

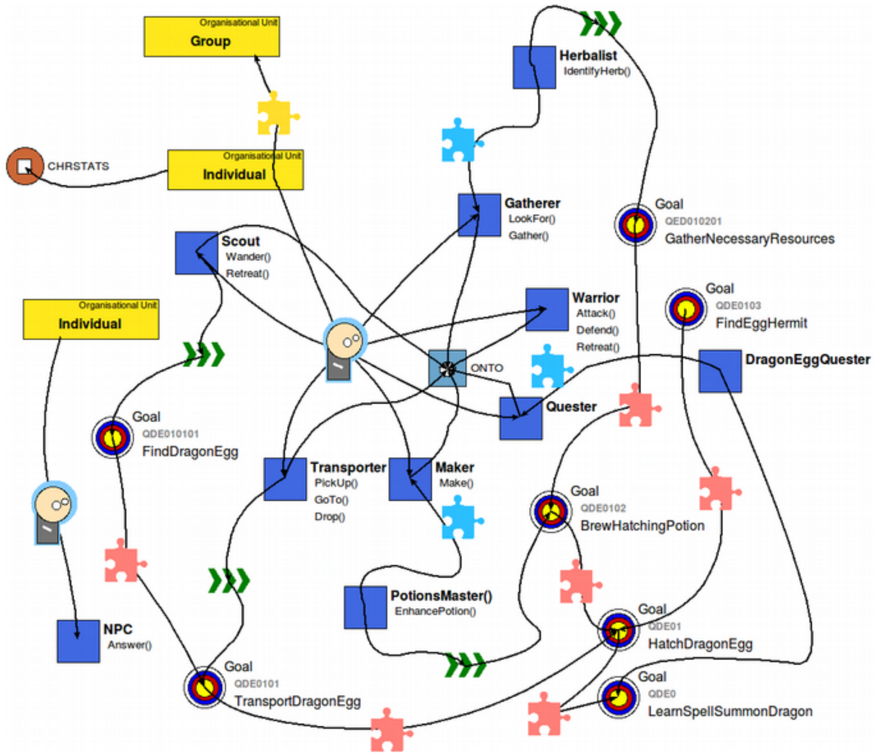

Fig. 5 Sample representation of the described Quest for the Dragon Egg with modelled Organisational Units, Roles, Knowledge Artefacts, Goals and Processes

In addition to visual elements, some concepts have attributes that can receive specific values. Goals have such attributes denoting means of measurement if a goal is satisfied, and rewards for completing the given goal. The goal named BrewHatchingPotion, shown in Fig. 5 has partially defined means of measuring if this particular goal is completed. The measurement is only partially defined in Fig. 6, since it does specify the necessary ingredients, but not the amount of these ingredients. Notice that both measurement and reward are given using a predicate logic expression. Therefore, reward predicate will be added to the main ontology upon finishing the goal, and measurement is based on already existing data. Depending on design decisions of the system developers, measurement can be based on data available from the world, an individual agent, or a combination of both, expressed in the desired way.

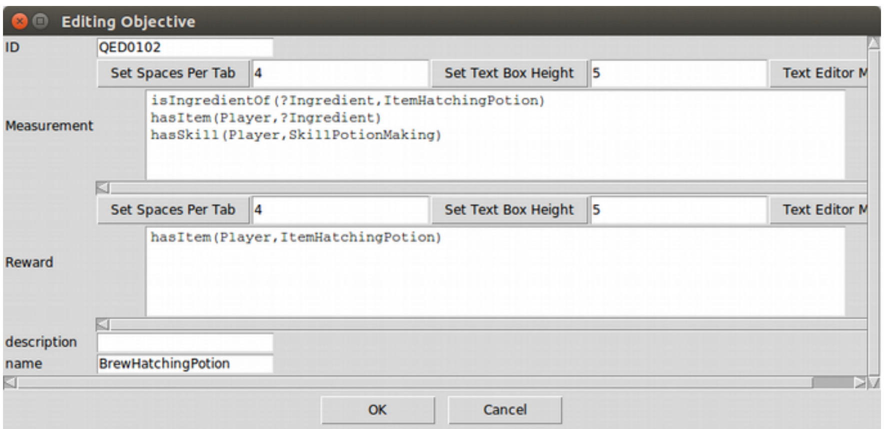

Fig. 6 Goal attributes and their values, where Measurement and Reward are motivated by the earlier mentioned seven perspectives of LSMAS organisational modelling

4 Discussion

This section provides a brief discussion on the modelled examples, covering arguments backing up the presented metamodel, along with guidelines suggesting further development of the metamodel.

The examples described above show what the modelling process' result may look like, and the amount of information it may include. Laying out plans of the resulting system is certainly a worthwhile step when one is planning a new system and development thereof.

Working with distributed systems of large scale can create a state of potential unreliability, if the system being developed is not documented well enough. The work-in-progress metamodel that was shown in specific examples here, is aiming to reduce the probability of such situations. Designed especially for modelling LSMAS, comprising concepts that may be used for that specific purpose, the metamodel offers system designers an easy and efficient way to lay down plans of their systems-to-be and analyse the assembled to-be model. Furthermore, it is possible to observe specific features that may inhibit usefulness, or success of the system being developed, and act accordingly.

From another perspective, metamodel showcased here can be used to facilitate easier or faster creation of a MAS specialised in distributed software project managements. The idea of an intelligent decision support and assisting system to be used by software project managers was laid out by Connor and Jenkins in [12].

Whichever of the above cases may be the prevalent one, managing distributed systems' development may prove to be easier and more efficient using the proposed metamodel, since not only does it provide an efficient overview of the modelled system, but it also makes generating basic computer code based on the defined model possible.

4.1 Evaluating the Proposed Approach to Modelling LSMAS

The proposed metamodel builds on recent studies of LSMAS organisational modelling, and includes several concepts that are not found in the models mentioned in Sect. 2 of this chapter. Novel as it is, since it is based on an ontology comprising concepts of organisational modelling of LSMAS, and it follows the recent trends in LSMAS development, it does have room for improvement.

Each of the modelled examples gives an insight about the metamodel from a specific perspective. The first example (Sect. 3.1) is about a simple system that requires no complex structures, therefore it shows how the metamodel can be applied to small-scale systems. Short analysis of this example yields the following conclusions.

The model is expressive enough to represent the described system in as much detail as is needed for clear description of the given system. Since the metamodel being developed allows for various levels of abstractness, the model of example one could have been even more simple.

The two modelled Organisational Unit concepts could have been merged into one, but the metamodel is not yet expressive enough to distinguish between logical AND and logical OR connections — it would be advisable to model the individual agent using only one Organisational Unit that has access to either of the Individual KnArts, and can play only one of the two modelled Roles at any given time.

It is clear from the built model that the system comprises agents playing two different roles, with no constraints on communication possibilities. Each of the individual agents have access to some individual knowledge, and every role knows the same set of organisational knowledge. The goal hierarchy is clear and easy to understand, though the process concept is lacking insofar as it is not known what actions of a role are included in the modelled specific process. Individual processes may as well be represented using another model using the metamodel being developed. Such an approach would further emphasize applicability of this metamodel to various levels of abstraction.

The second example, on the other hand, shows what a rather more complex system looks like being modelled using the proposed metamodel. Although a rather complex situation (that is a part of a larger world), it is easily and clearly modelled. Short analysis of this example yields the following conclusions.

An individual agent, modelled using an Organisational Unit concept, is defined using a slightly different approach from the one in example one. Only two types of individual agents are planned — agent commanded by a player, and agent playing the role of a Non Player Character (NPC). Player avatars, as individual agents controlled by a player, can be grouped into a party (a group of player avatars). Furthermore, every individual avatar has some basic stats, and their own inventory, as is shown using individual knowledge artefacts.

Diversity of roles is obvious, and their inclusion structure is clearly defined. The property denoting one role as a part of another role can be understood as an inheritance property, thus actions defined for roles on higher level (e.g. Gatherer, Maker, Warrior, etc.) are inherited by roles on lower levels (e.g. Herbalist, PotionsMaster, SupportWarrior, etc.). It is therefore concluded that an organisational unit that can access a higher level role can also access a lower level role.

The main goal of the questing role is decomposed on several subgoals. As is visible from the model, the main goal structure is not completely linear, i.e. subgoal QDE01 is decomposed on three different goals with possible subgoals (e.g. QDE0101, QDE0102, QDE0103).

The model further shows that some of the subgoals can only be finished by a specific role (e.g. QED0102 named BrewHatchingPotion can only be finished by role named PotionsMaster). Naturally, every role has further constraints on when it can be played by a certain individual agent, but those constraints are not described in this particular model, nor at this stage of the metamodel development.

Reusability of concepts, as a feature of the proposed metamodel is shown using various concepts of the model, yet it is most visible in joining roles to organisational knowledge artefacts. Two organisational knowledge artefacts, one representing the domain ontology, and the other ontology comprising communication concepts, are modelled only once, and are used by many different properties. Furthermore, roles do not have to be modelled more than once, but can be used by several different organisational units, and by various properties.

The metamodel, as work-in-progress, can be evaluated as follows. The model is lacking in constraints of playing roles — it would seem that any individual agent may play any role at any given time. Although such a presumption may be true in the modelled part of the bigger system, it may be necessary to add constraint possibilities, e.g. implementing logic AND and OR expressions. This type of connection that would denote partial or complete inclusion of the concepts of property range may be useful in properties denoting hierarchical inclusion of concepts.

The two examples show how applicable the metamodel can be in situations of different size containing various elements. Even though the metamodel can be used for modelling the simple example, it is expressive enough to model the more complex example as well. The obvious problem in modelling the examples of this chapter is cluttered view, and chaotic placement of numerous elements of the model. This is largely due to the very early stage of development of the model, since one of the features offered by the finished model is capability of modelling various levels of abstraction in different layers of the model, thus removing the visible clutter.

Since the metamodel is a work in progress, some planned features have not yet been implemented, wherefore is does not completely comply with the modern trends of LSMAS modelling, i.e. perspectives of LSMAS modelling, as noted earlier, in Sect. 2.2. A notable feature missing is modelling of interorganisational dynamics, i.e. concepts for modelling the mentioned aspect of organisation. An introduction towards modelling organisations as individual agents is shown in example two, where there exists an organisational unit representing a group of individual agents. Interaction between such organisational units that represent groups may be modelled in the same layer, or level of abstraction, as the one shown in example two, or in another one, represented by a new model.

The model, in its current state, partially or completely satisfies the following perspectives of LSMAS modelling mentioned in Sect. 2.2: organisational structure (since it supports modelling of decision and information flows), organisational culture (knowledge is modelled on individual or organisational level, norms are implemented using role concepts, and language is supported as a knowledge artefact, while goals do provide rewards upon being successfully completed), strategy (it is possible to model goals or objectives, and their subgoals, i.e. how they relate to other goals, and how goal success is measured), processes (every role can have defined activities it can perform, and those activities can be combined into a process that has a specific goal), and individual agents (it is possible to model individual agents insofar as to designate they exist, what knowledge they possess, and what roles an individual can play). As mentioned earlier, organisational dynamics, and context and interorganisational aspects have not been defined yet, although the first elements of these two perspectives are visible.

5 Conclusions

This chapter is about modelling LSMAS that conform to various elements of organisational design. A relevant set of perspectives of organisational architecture that is proposed to be used for modelling modern LSMAS was presented in a study by Schatten [47]. The work in progress metamodel presented in this chapter is based on the said set of perspectives, and aims to provide a relevant upgrade of some recent studies of LSMAS organisational modelling.

The model presented in this chapter is presented as a suitable tool for planning and modelling LSMAS, in their many application domains, ranging from MMOGs, to smart cities, smart transport, and smart infrastructure, to distributed systems in general. Since planning and modelling phases have a great impact on the rest of the lifecycle of a system, it is argued to be interesting to use a tool that allows one to create a model of the system being built, and generate basis of the said system upon the defined model, which is a feature the metamodel presented in this chapter is intended to provide.

Outlined by the overview given in Sect. 2.1, the proposed metamodel should be expressive enough, yet simple to use, to provide the user with concepts that can model a simple example, as well as a more complex one. One of the main features of the proposed metamodel are recursive definitions of organisational units, goals and roles, as represented formally by Schatten [47]. Such an approach was shown in second example (Sect. 3.2) where an organisational unit represents both an individual agent, and a group of agents. Further examples are used for modelling goals and roles in Sect. 3.2 as well.

The author argues that using the proposed metamodel (once finished) may be beneficial for planning and modelling phases of development of distributed systems, as it allows the system developers to model the system in enough detail to represent functionality of the system in a way that is not overly complex, and that is easy to read and comprehend. Therefore, although currently in development, and lacking in features, the proposed metamodel is argued to be a useful addition to recent research on LSMAS organisational modelling.

Future work concerning the proposed metamodel is clearly designated by the features lacking when compared to the perspectives of LSMAS organisational architecture used as the starting idea behind this metamodel. Further research into the existing models is needed, in form of a more thorough overview or analysis of those most recent, so as to compare the metamodel being developed (once finished) to those analysed. Based on the ontology pertaining selected organisational concepts for organisation of LSMAS, concepts included in the metamodel at this stage have to be analysed as well, in order to determine if some of the included concepts are redundant, or simply not needed, and there are concepts that have to be added. Furthermore, additional specific scenarios from an LSMAS application domain will be identified and the developed metamodel tested on them, so as to identify weak elements of the metamodel, and needed improvements.

7 Acknowledgements

This work has been supported in full by the Croatian Science Foundation under the project number 8537.

References

1. Abbas HA (2014) Exploiting the Overlapping of Higher Order. Int J Agent Technol Syst 6:32–57. doi: 10.4018/ijats.2014070102
2. Abbas HA (2015) Realizing the NOSHAPE MAS Organizational Model: Int J Agent Technol Syst 7:75–104. doi: 10.4018/IJATS.2015040103
3. Abbas HA, Shaheen SI, Amin MH (2015) Organization of Multi-Agent Systems: An Overview. Int J Intell Inf Syst 4:46–57. doi: 10.11648/j.ijiis.20150403.11
4. Atzori L, Iera A, Morabito G (2010) The Internet of Things: A survey. Comput Networks 54:2787–2805. doi: 10.1016/j.comnet.2010.05.010
5. Bădică A, Bădică C, Ganzha M, Ivanović M, Paprzycki M (2016) Experiments with Multiple BDI Agents with Dynamic Learning Capabili - ties. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer

International Publishing, Cham, pp 274–286

6. Barriuso AL, de La Prieta F, Murciego ÁL, Hernández D, Herrero JR (2016) An Intelligent Agent-Based Journalism Platform. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 322–332
7. Bergenti F, Iotti E, Poggi A (2016) Core Features of an Agent-Oriented Domain-Specific Language for JADE Agents. In: de la Prieta F, Escalona JM, Corchuelo R, Mathieu P, Vale Z, Campbell TA, Rossi S, Adam E, Jiménez-López DM, Navarro ME, Moreno NM (eds) Trends Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Springer International Publishing, Cham, pp 213–224
8. Boomsma JJ, Franks NR (2006) Social insects: from selfish genes to self organisation and beyond. Trends Ecol Evol 21:303–308. doi: 10.1016/j.tree.2006.04.001
9. Boulaire F, Utting M, Drogemuller R (2015) Dynamic agent composition for large-scale agent-based models. Complex Adapt Syst Model 3:1–23. doi: 10.1186/s40294-015-0007-2
10. Van Den Broek EL, Jonker CM, Sharpanskykh A, Treur J, others (2006) Formal modeling and analysis of organizations. In: Boissier O, Padget J, Dignum V, Lindemann G, Matson E, Ossowski S, Sichman JS, Vázquez-Salceda J (eds) Coord. Organ. Institutions, Norms Multi-Agent Syst. Springer Berlin Heidelberg, pp 18–34
11. Bui N, Zorzi M (2011) Health care applications: a solution based on the internet of things. In: Proc. 4th Int. Symp. Appl. Sci. Biomed. Commun. Technol. - ISABEL '11. ACM Press, New York, New York, USA, pp 1–5
12. Connor RO, Jenkins J Using Agents for Distributed Software Project Management. Management
13. Corkill DD, Lander SE (1998) Diversity in agent organizations. Object Mag 8:41–47.
14. Coutinho LR, Sichman JS, Boissier O (2009) Modelling Dimensions for Agent Organizations. In: Dignum V (ed) Handb. Res. Multi-Agent Syst. IGI Global, pp 18–50
15. Čyras K (2016) Argumentation-Based Reasoning with Preferences. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 199–210
16. Decker KS (1996) TÆMS : A Framework for Environment Centered Analysis & Design of Coordination Mechanisms. In: Found. Distrib. Artif. Intell. John Wiley & Sons, Inc, pp 429–448
17. Dignum V (2004) A model for organizational interaction: based on agents, founded in logic. Utrecht University
18. Van Dyke Parunak H, Brueckner S (2001) Entropy and Self-Organization in Multi-Agent Systems. In: Proc. Int. Conf. Auton. Agents. Montreal, Canada, pp 124–130
19. Van Dyke Parunak H, Odell J (2001) Representing social structures in UML. In: Proc. fifth Int. Conf. Auton. agents - AGENTS '01. ACM Press, New York, New York, USA, pp 100–101
20. Esteva M, Padget J, Sierra C (2002) Formalizing a Language for Institutions and Norms. In: Meyer J-JC, Tambe M (eds). Springer Berlin Heidelberg, Berlin, Heidelberg, pp 348–366
21. Fabretti A, Gärling T, Herzel S, Holmen M (2016) An Agent-Based Model to Study the Impact of Convex Incentives on Financial Markets. In: Trends Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. pp 3–13
22. Ferber J, Gutknecht O, Michel F (2004) From agents to organisations: an organizational view of multi-agent systems. Agent-Oriented Softw Eng IV 214–230. doi: 10.1007/978-3-540-24620-6_15
23. Fontana M, Terna P (2015) From Agent-based models to network analysis (and return): the policy-making perspective. Work. Pap. Ser. 7:
24. Garcia-Rodriguez S, Sleiman HA, Nguyen V-Q-A (2016) A Multi-agent System Architecture for Microgrid Management. In: de la Prieta F, Escalona JM, Corchuelo R, Mathieu P, Vale Z, Campbell TA, Rossi S, Adam E, Jiménez-López DM, Navarro ME, Moreno NM (eds) Trends Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Springer International Publishing, Cham, pp 55–67
25. Gasser L (2001) Perspectives on Organizations in Multi-agent Systems. In: Luck M, Mařík V, Štěpánková O, Trappl R (eds) Multi-Agent Syst. Appl. Springer Berlin Heidelberg, Berlin, pp 1–16
26. Gliwa B, Koźlak J, Zygmunt A, Demazeau Y (2016) Combining Agent-Based and Social Network Analysis Approaches to Recognition of Role Influence in Social Media. In: Demazeau Y, Ito T, Bajo J, Escalona JM (eds) Adv. Pract. Appl. Scalable Multi-agent Syst. PAAMS Collect. 14th Int. Conf. PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proc. Springer International Publishing, Cham, pp 109–120
27. Hadfi R, Ito T (2016) Holonic Multiagent Simulation of Complex Adaptive Systems. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 137–147
28. Hernández D, Villarrubia G, Barriuso AL, Lozano Á, Revuelta J, De Paz JF (2016) Multi Agent Application for Chronic Patients: Monitoring and Detection of Remote Anomalous Situations. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 27–36
29. Horling B, Lesser V (2005) A survey of multi-agent organizational paradigms. Knowl Eng Rev 19:281. doi: 10.1017/S0269888905000317
30. Hübner JF, Sichman JS, Boissier O (2002) A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In: Bittencourt G, Ramalho GL (eds) Adv. Artif. Intell. Springer Berlin Heidelberg, pp 118–128
31. Hübner JF, Vercouter L, Boissier O (2009) Instrumenting Multi-agent Organisations with Artifacts to Support Reputation Processes. In: Hübner JF, Matson E, Boissier O, Dignum V (eds) Coord. Organ. Institutions Norms Agent Syst. IV. Springer Berlin Heidelberg, Berlin,

Heidelberg, pp 96–110

32. Kir H, Erdo\ugan N (2016) Agent-Based Semantic Business Process Management Methodology. In: Demazeau Y, Ito T, Bajo J, Escalona JM (eds) Adv. Pract. Appl. Scalable Multi-agent Syst. PAAMS Collect. 14th Int. Conf. PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proc. Springer International Publishing, Cham, pp 145–156
33. Lofgren ET, Fefferman NH (2007) The untapped potential of virtual game worlds to shed light on real world epidemics. Lancet Infect Dis 7:625–629. doi: 10.1016/S1473-3099(07)70212-8
34. Losilla J, Olivares T, Fernández-Caballero A (2016) Multi-agent-Based Framework for Prevention of Violence Against Women: Scenarios in Google Maps. In: de la Prieta F, Escalona JM, Corchuelo R, Mathieu P, Vale Z, Campbell TA, Rossi S, Adam E, Jiménez-López DM, Navarro ME, Moreno NM (eds) Trends Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Springer International Publishing, Cham, pp 277–285
35. McCauley L, Franklin S (2002) A large-scale multi-agent system for navy personnel distribution. Conn Sci 14:371–385. doi: 10.1080/0954009021000068934
36. Mihaylov M, Razo-Zapata I, R\uadulescu R, Jurado S, Avellana N, Nowé A (2016) Smart Grid Demonstration Platform for Renewable Energy Exchange. In: Demazeau Y, Ito T, Bajo J, Escalona JM (eds) Adv. Pract. Appl. Scalable Multi-agent Syst. PAAMS Collect. 14th Int. Conf. PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proc. Springer International Publishing, Cham, pp 277–280
37. Okreša Đurić B (2016) Organizational Metamodel for Large-Scale Multi-Agent Systems. In: de la Prieta F, Escalona MJ, Corchuelo R, Mathieu P, Vale Z, Campbell AT, Rossi S, Adam E, Jiménez-López MD, Navarro EM, Moreno MN (eds) Adv. Intell. Syst. Comput. Springer International Publishing, Seville, ES, pp 387–390
38. Okreša Đurić B, Konecki M (2015) Modeling MMORPG Players' Behaviour. In: Hunjak T, Kirinić V, Konecki M (eds) Cent. Eur. Conf. Inf. Intell. Syst. 2015. Faculty of Organization and Informatics, Varaždin, HR, pp 177–184
39. Okreša Đurić B, Schatten M (2016) Defining Ontology Combining Concepts of Massive Multi-Player Online Role Playing Games and Organization of Large-Scale Multi-Agent Systems. 39th Int. Conv. Inf. Commun. Technol. Electron. Microelectron.
40. Parrish JK, Viscido S V., Grünbaum D (2002) Self-Organized Fish Schools: An Examination of Emergent Properties. Biol Bull 202:296–305. doi: 10.1016/j.foodchem.2006.01.008
41. Picard G, Hübner JF, Boissier O, Gleizes M-P (2009) Reorganisation and Self-organisation in Multi-Agent Systems. In: Int. Work. Organ. Model. Paris, FR, pp 66–80
42. Pico-Valencia P, Holgado-Terriza JA (2016) ADELE: A Middleware for Supporting the Evolution of Multi-agents Systems Based on a Metaprogramming Approach. In: de la Prieta F, Escalona JM, Corchuelo R, Mathieu P, Vale Z, Campbell TA, Rossi S, Adam E, Jiménez-López DM, Navarro ME, Moreno NM (eds) Trends Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Springer International Publishing, Cham, pp 297–310
43. Posey RB, Haire M (1961) Modern Organization Theory. Adm Sci Q 5:609–611. doi: 10.2307/2390625
44. Román JA, Rodríguez S, de la Prieta F (2016) Improving the Distribution of Services in MAS. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 37–46
45. Russell SJ, Norvig P (2010) Artificial Intelligence: A modern Approach, 3rd ed. Prentice Hall, New Jersey, USA
46. Schatten M (2012) Complex Analytical Method for Self-organizing Multiagent Systems. In: Cent. Eur. Conf. Inf. Intell. Syst. 2012. Faculty of Organization and Informatics, Varaždin, HR, pp 63–70
47. Schatten M (2014) Organizational Architectures for Large-Scale Multi-Agent Systems' Development: An Initial Ontology. Adv Intell Syst Comput 290:261–268. doi: 10.1007/978-3-319-07593-8_31
48. Schatten M, Ševa J, Tomičić I (2016) A roadmap for scalable agent organizations in the Internet of Everything. J Syst Softw 115:31–41. doi: 10.1016/j.jss.2016.01.022
49. Schatten M, Tomicic I, Okreša Đurić B (2015) Multi-agent modeling methods for massivley Multi-Player On-Line Role-Playing Games. In: Biljanović P (ed) 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. IEEE, Opatija, HR, pp 1256–1261
50. Scheutz M (2010) A Multi-Agent System Infrastructure for Large-Scale Autonomous Distributed Real-Time Intelligence Gathering Systems. Proc. ISCA
51. Tomičić I (2016) Agent-based Framework for Modelling and Simulation of Resource Management in Smart Self-Sustainable Human Settlements. University of Zagreb
52. Tomičić I, Schatten M (2015) Towards an Agent Based Framework for Modelling Smart Self-Sustainable Systems. Interdiscip Descr Complex Syst 13:50–63. doi: 10.7906/indecs.13.1.7
53. Tomičić I, Schatten M (2016) Agent-based framework for modeling and simulation of resources in self-sustainable human settlements: a case study on water management in an eco-village community in Croatia. Int J Sustain Dev World Ecol 1–10. doi: 10.1080/13504509.2016.1153527
54. Tsarev A, Skobelev P (2016) Multi-agent Supply Scheduling System Prototype for Energy Production and Distribution. In: Demazeau Y, Ito T, Bajo J, Escalona JM (eds) Adv. Pract. Appl. Scalable Multi-agent Syst. PAAMS Collect. 14th Int. Conf. PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proc. Springer International Publishing, Cham, pp 290–293
55. Vermesan O, Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Bassi A, Jubert IS, Mazura M, Harrison M, Eisenhauer M, Doody P, Peter F, Patrick G, Sergio G, Harald, Sundmaeker Alessandro B, Ignacio Soler J, Margaretha M, Mark H, Markus E, Pat D (2009) Internet of Things Strategic Research Roadmap. In: Vermesan O, Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Bassi A, Jubert IS (eds) Internet

Things Strateg. Res. Roadmap. pp 9–52

56. Villavicencio C, Schiaffino S, Diaz-Pace JA, Monteserin A, Demazeau Y, Adam C (2016) A MAS Approach for Group Recommendation Based on Negotiation Techniques. In: Demazeau Y, Ito T, Bajo J, Escalona JM (eds) Adv. Pract. Appl. Scalable Multi-agent Syst. PAAMS Collect. 14th Int. Conf. PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Proc. Springer International Publishing, Cham, pp 219–231
57. Vlacheas P, Giaffreda R, Stavroulaki V, Kelaidonis D, Foteinos V, Poulios G, Demestichas P, Somov A, Biswas A, Moessner K (2013) Enabling smart cities through a cognitive management framework for the internet of things. IEEE Commun Mag 51:102–111. doi: 10.1109/MCOM.2013.6525602
58. Weyns D, Haesevoets R, Helleboogh A (2010) The MACODO organization model for context-driven dynamic agent organizations. ACM Trans Auton Adapt Syst 5:1–29. doi: 10.1145/1867713.1867717
59. Weyns D, Haesevoets R, Helleboogh A, Holvoet T, Joosen W (2010) The MACODO middleware for context-driven dynamic agent organizations. ACM Trans Auton Adapt Syst 5:1–28. doi: 10.1145/1671948.1671951
60. Wikarek J, Sitek P (2016) A Multi-level and Multi-agent Approach to Modeling and Solving Supply Chain Problems. In: Bajo J, Escalona JM, Giroux S, Hoffa-D\kabrowska P, Julián V, Novais P, Sánchez-Pi N, Unland R, Azambuja-Silveira R (eds) Highlights Pract. Appl. Scalable Multi-Agent Syst. PAAMS Collect. Int. Work. PAAMS 2016, Sevilla, Spain, June 1-3, 2016. Proc. Springer International Publishing, Cham, pp 49–60
61. Yu H, Shen Z, Leung C (2013) From Internet of Things to Internet of Agents. In: 2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput. IEEE, pp 1054–1057
62. Žugaj M, Schatten M (2005) Arhitektura suvremenih organizacija. Tonimir i Fakultet organizacije i informatike, Varaždinske Toplice
63. Žugaj M, Šehanović J, Cingula M (2004) Organizacija, 2nd ed. TIVA Tiskara Varaždin, Varaždin, HR

Index