



ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal  
 Regular Issue, Vol. 6 N. 3 (2017), 17 -27  
 eISSN: 2255-2863  
 DOI: [http://dx.doi.org/10.14201/ADCAIJ20176317\\_27](http://dx.doi.org/10.14201/ADCAIJ20176317_27)

# Organisational Metamodel for Large-Scale Multi-Agent Systems: First Steps Towards Modelling Organisation Dynamics

Bogdan Okreša Đurić

Faculty of Organization and Informatics, University of Zagreb, Pavlinska 2, 42000 Varaždin, Croatia  
 dokresa@foi.hr

## KEYWORD

*Organisation;  
 Metamodel;  
 Large-scale  
 Multiagent  
 Systems;  
 Distributed  
 Systems;  
 Dynamics*

## ABSTRACT

*The research presented in this paper is a thesis proposal with the main goal of defining an ontology comprising chosen organisational concepts applicable to large-scale multiagent systems (LSMAS), and building a metamodel for modelling selected organisational features in such systems. The method of applying aspects of human organisations to multiagent systems (MAS) comprising autonomous intelligent agents will be enriched through this research with a new perspective of modelling organisation dynamics in LSMAS. Results of this research, in their final version, will be tested using testbed scenarios based on a specific massively multi-player online role-playing game (MMORPG), since MMORPGs are one of the identified application domains of LSMAS. It is important to note that results described in this paper showcase partial results in their early stage of development. Nevertheless, first traces of a modelling tool that is expected to aid in development of LSMAS for numerous application domains, and ease their organisational design, are recognisable in the proposed combination of ontology engineering, metamodeling and code generating methods.*

## 1. Introduction

This research stems from the ModelMMORPG project<sup>1</sup> with main goals of identifying and formalizing adequate organizational design methods for developing LSMAS in MMORPGs, coupling them with real-life and future scenarios from industry, and providing open and accessible tools that will allow for design, development, implementation, control, simulation and maintenance of LSMAS in MMORPGs.

The research described in this paper features and showcases first steps towards the novel organisational metamodel for design and development of LSMAS. The most prominent feature of the finished (meta)model will be modelling of organisation dynamics. Significance of organisation dynamics is derived from the basic elements of the ModelMMORPG project and its work on MMORPGs: more often than not, players in MMORPGs are

1. Large-Scale Multi-Agent Modelling of Massively On-Line Role-Playing Games, conducted by AI Lab of the Faculty of Organization and Informatics, University of Zagreb, Croatia, <http://ai.foi.hr/modelmmorpg.php>.



demanding to form coalitions (in a game usually called parties when featuring a couple of players, or e.g. guilds when featuring dozens of players). These coalitions are not permanent, and can change in very short intervals. E.g. a party with specific organisational features may be forced into introducing changes to its structure or another aspect of organisation, driven by death of one of the units of this organisation (an avatar, i.e. an in-game protagonist character), by players dropping out of the game, etc. Therefore, managing changes in such coalitions, that can be observed as organisations for their basic organisational features, is a challenge to be reckoned with. Furthermore, such a behaviour of agents, i.e. coalition forming, may be noticed not exclusively in MMORPGs, but in a wide variety of MAS and LSMAS application domains as well. The mentioned metamodel will be based on an ontology comprising chosen organisational concepts that can be applied to LSMAS, since a good foundation upon which the mentioned metamodel will be built allows for easier introduction of further improvements into the metamodel, once it is finished.

MMORPGs are an interesting domain to be researched when behaviour of agents is observed. Millions of players (Schatten *et al.*, 2015) that spend countless hours on playing MMORPGs are a valuable source of data and information on how people act and organise in unexpected situations (Lofgren and Fefferman, 2007), although presumably without fatal consequences, and how they act when confronted with tasks that demand cooperation (Schatten and Okreša Đurić, 2015). It may be easily understood why massively multi-player online games (MMOGs) in general are a good source of information, since all the communication between the included agents, be it real-life humans or artificial agents, is happening digitally, and is easily analysed. MMORPGs (Schatten and Okreša Đurić, 2016) are a mix of role-playing games (RPGs) and MMOGs allowing players to control actions of their avatar simultaneously interacting with a potentially large user-base online in a well defined common world. The stated can be abstracted nicely using LSMAS, and is therefore a good application domain example. Core elements of MMORPGs can be successfully applied to other domains as well, since they represent basic elements of various interactions (avatar characteristics, tasks, a range of items, etc.).

Applying features of human organisations on MAS is not a new concept per se. Indeed, various approaches to organising MAS were conducted in the last decade alone. Most notable approaches to organisational features in MAS stem from two extremes in observing organisation (De Wolf, 2004): as an entity, or as a process. The concept of organisation in MAS is usually observed using the following two distinct concepts (Abbas *et al.*, 2015): agent-centred multiagent systems (ACMAS) and organisation-centred multiagent systems (OCMAS). The idea of organisation in ACMAS is not something that can be cast upon the system by force. On the contrary, it emerges from agents interacting and affecting the environment (including other agents) through their actions, behaviour and interaction in general. The most prominent examples of ACMAS include insect swarms and colonies of ants (De Wolf, 2004; Abbas *et al.*, 2015). On the other hand, organisation in OCMAS is considered as a concept consisting of rules, and clearly defined information and decision flows, enforced upon the system, although the agents included can affect the set rules if they recognise a change is needed in order to adapt to the unstable environment. It is argued that OCMAS approach is more suitable for complex systems, although systems functioning in highly turbulent environments benefit the most from a combination of both approaches (Corkill and Lander, 1998; Dignum, 2009). Turbulence is one of the main features of modern environments where MAS and LSMAS should be situated, as for example in the Internet of Everything (IoE) (Schatten *et al.*, 2016).

The rest of the paper is structured as follows. A brief overview of research goals is given in section 2. The used methodology is described in section 3. Since some research has been done already, available partial results are presented in section 4, followed by an example featuring elementary characteristics of the proposed metamodel in section 5. Finally, the shown example and the present state of the metamodel being developed will be critically observed in section 6. The paper will be concluded with further research steps, and an overview of the presented in section 7.

## 2. Goals of the Research

Two main goals of this research are to define an ontology comprising chosen organisational concepts applicable to LSMAS, and to build an organisation metamodel for LSMAS based on the mentioned ontology (Okreša Đurić,

2016), both conforming to the seven perspectives of organisation modelling defined by (Schatten, 2014), but emphasising organisational dynamics.

The mentioned perspectives of organisation modelling represent modern basis for modelling LSMAS with organisational features, ready to cope with increasingly demanding environment of the modern world. Detailed in (Schatten, 2014), there are seven perspectives: organisational structure, organisational culture, strategy, processes, individual agents, organisational dynamics, and finally context and inter-organisational aspects. The mentioned perspectives are tightly connected to organisational features of human organisations, but are presented as well-adapted for LSMAS.

Based on the main goal of this research, three objectives have been identified, as follows:

- O1 Analyse organisational modelling concepts applicable to LSMAS.
- O2 Model organisational concepts applicable to MMORPG.
- O3 Explore modelling of organisational dynamics in MMORPG as specific LSMAS application.

The stated research objectives are aligned with the main goals of this research through their outputs. Analysis of organisational modelling concepts applicable to LSMAS shall yield an ontology of organisational concepts that can be used to describe organisational features of an LSMAS. This ontology shall be used as input for the process of achieving the other two objectives. Methodology that is to be used in the research process is described in section 3 of this paper. Conformance with the organisational perspectives mentioned above is dealt with in steps of the methodology considered suitable for this research. The stated research objectives are clearly oriented towards the world of MMORPGs, but they will yield results that can be applied to other LSMAS application domains as well. In addition to methodological frame of this research, section 3 of this paper will describe in more detail how results will be tested against the objectives above.

### 3. Proposed Methodology

The objectives set in section 2 define the methodology to be used for this research. Since these objectives implicitly include ontology design and metamodelling processes, applicable methodologies are described here, and since the metamodel will have to be verified, a proposed testing scenario is described as well.

#### 3.1. Ontology Design

Upon analysing ontology engineering methodologies presented in (Iqbal *et al.*, 2013), a certain ontology engineering methodology was chosen as the most appropriate one for the intended research. METHONTOLOGY (Fernández-López *et al.*, 1997) is based on a developing prototype, has reusability support, is not dependent on a specific application environment, is very well described, and has a clear ontology life cycle recommendation. Steps identified based on the available description of METHONTOLOGY are as follows.

The first step is a combination of two simultaneous activities: specification, and knowledge acquisition. While a document containing basic information about the ontology being developed is the outcome of the specification activity, knowledge acquisition is about finding and analysing existent potential sources of the concepts the ontology being developed is aimed at. Following the defined specification document, an index comprising chosen and identified concepts with their definitions should be built during the conceptualisation step. Since one of the main goals of an ontology is knowledge sharing, the following step, integration, is aimed at connecting the ontology being developed with existing ontologies and metaontologies. The ontology is completely defined during the implementation step using the ontology development environment Protégé. The last step is aimed at verification, validation, evaluation, and technical judgement of the developed ontology, with respect to a frame of reference (usually the output of the specification step mentioned earlier).

Two interesting sources of concepts to be included in the final form of the ontology being developed were identified in the ontology from OOVASIS project<sup>2</sup> (containing necessary organisational concepts) (Schatten *et al.*, 2014), and MAM5 ontology (containing concepts for defining an intelligent virtual environment) (Barella *et al.*, 2012). The stated ontologies are chosen on the basis of both of them being recent, and closely related to LSMAS. Furthermore, a possible combination of the mentioned ontologies is believed to represent a useful basis for the desired metamodel development. Ontology development environment Protégé was chosen because it is open-source, very popular in both academic and real sectors, and it is a widely accepted tool for ontology engineering. The ontology will be defined using Web Ontology Language (OWL) detailed in (W3C OWL Working Group, 2012) because of its role of a de facto standard for describing ontologies. OWL is used to introduce concepts enhancing expressiveness of Resource Description Framework (RDF) (Schreiber *et al.*, 2014), which in turn is a framework for expressing information about resources.

A part of the novelty presented by this research will be visible in the ontology part of research outcome, since it has not come to the attention of the author that LSMAS concepts have been intertwined with organisational concepts on such a level before. The key contribution of this merging will be concepts that can be used for modelling organisational dynamics, since that is an important part of LSMAS, as will be demonstrated on a specific MMORPG scenario. An interesting feature of the OOVASIS ontology is recursive definition of the included concepts. For example, an organisation unit is informally defined as an individual agent that is a part of an organisation, but it can represent an organisation built of individual agents as well, which is an idea similar to the concept of holons (Van Dyke Parunak and Odell, 2001). This approach is applied to some other organisation features as well, such as organisational processes, strategy, and knowledge artefacts (Schatten, 2014). One of the main ideas of this research, and, consequently, the ontology and the metamodel being developed, is to retain this recursive approach to various chosen and included concepts, for the sake of a more generally applicable results.

### 3.2. Building and Testing the Metamodel

After the ontology is defined, the model is to be developed. The scientific model developing method consists of several phases (Žugaj, 2007): setting a goal, building a model, detailing the model, and applying the model. Following these phases, five steps were identified: defining the level of abstractness of the metamodel; choosing the appropriate concepts from the developed ontology; analysing and assessing LSMAS organisational models that exist already; developing the metamodel; and assessing the developed metamodel.

The specific situation of metamodelling part of this research arises from the fact that the metamodel being developed is highly dependent on the ontology being developed. Therefore, constant work should be done on both of these elements, demanding customisation of one according to the other, and vice versa. Metamodelling step has already been commenced using AToM<sup>3</sup> (de Lara and Vangheluwe, 2002) since it is an open-source software working with programming language Python, and is by default built for meta- and modelling purposes. Furthermore, it makes it possible to graphically create and use a model, and to introduce numerous modifications to it. Easy integration with Python is important since a clear and efficient platform for development of MAS (SPADE, Smart Python multi-Agent Development Environment) (Gregori *et al.*, 2006) was developed using Python, and Python provides many libraries of various possible applications. SPADE is the first such piece of software ever to use a particular popular communication protocol (XMPP).

The final version of the metamodel being developed should provide a feature for modelling organisation dynamics, motivated in section 1 of this paper. A presumably good choice of methods for modelling organisation dynamics in a satisfiable way includes graph grammars, and temporal logics. Since the metamodel will be, and is in its current state, represented as a graph, with vertices showing various concepts of the modelled system connected by edges, graph grammars are a logical step towards introducing possible predefined changes to such graphs. On account of change being usually perceived as a temporal event, it is necessary to introduce temporal component in the model being developed, a feature which will most probably be utilised using temporal logics.

---

2. Organizational Design of Multi-Agent Systems in the Internet of Things (cro. Organizacijsko oblikovanje višag, agentnih sustava u Internetu Stvari), finished in June 2014 by AI Lab of the Faculty of Organization and Informatics, University of Zagreb, Croatia, <http://ai.foi.hr/ooovasis.php>.

The final version of the metamodel is envisioned as a tool for layered modelling of LSMAS, e.g. modelling agents on one layer, normative elements (e.g. groups of norms modelled as roles) on another, strategic elements (e.g. tasks and goals) on a different layer, etc. Such an approach should expedite LSMAS modelling process. Furthermore, the final version of the metamodel being developed should be able to generate a skeleton of the modelled system using concepts of SPADE environment. The generated skeleton will then be open to further development of the system's developer.

Once finished, the metamodel will be tested using testbed scenarios created as a part of the ModelMMORPG project mentioned earlier. The tests will involve organisations modelled using the developed metamodel, and intelligent agents being developed as part of the mentioned project (Maliković and Schatten, 2015), situated in an MMORPG environment. As mentioned earlier, MMORPGs are considered good examples of LSMAS application domain. Therefore, a specific MMORPG containing all the elements of a standard game of the mentioned genre, The Mana World, is chosen for the environment of the proposed test (Schatten *et al.*, 2015). Furthermore, the metamodel must conform to the perspectives of organisation modelling of LSMAS mentioned in section 2 of this paper.

## 4. Present Results

Some of the basic elements present in the OOVASIS ontology, envisioned to be included in the final version of the ontology being developed during this research, have already been included in their most basic form into the metamodel being developed, and a basic code generating ability has been implemented, as a conceptual proof of possible achieving of the desired results. The metamodel contains neither the wanted layered construction, nor any organisation dynamics concepts at this point in development though. The concepts that are currently available in the metamodel are: an organisation unit, a role, a goal, a process, individual and organisation knowledge artefacts, role inheritance, aggregation of organisation units, goal specification (goals and subgoals), and various properties between the mentioned concepts.

In this early stage of development of the metamodel, a simple system may be modelled using the mentioned elements. Furthermore, code generating feature is present as well, although it will be subject to further development. The metamodel does feature some organisational features, although any structure related concepts are, at the moment, constrained to defining control flows using hierarchical properties.

Code generating feature in its current form helps in generating code necessary for creating basic elements needed for agent creation using SPADE. At the present, fundamental elements of an agent can be defined, along with the requirements for running one instance of each of the actual organisation units modelled. Furthermore, only behaviours of modelled roles are generated at the moment, since role playing is envisioned as a matter of change. Indeed, an organisation unit is meant to start almost as a *tabula rasa* individual, and most of the actions, behaviour, and knowledge pertaining to the system, which it is a part of, should be received through playing various roles. A special agent should be present in the modelled system, one that will be playing a role of a special kind of teacher that knows which roles are viable for specific situations in which organisation units find themselves.

Essential programming code of a SPADE agent's behaviour (named NiceBehaviour simply printing out text designated at line 4 when enacted) is shown in listing 1, and key code elements of a SPADE agent (named Wizard, with one attached behaviour) are shown in listing 2. Use of the mentioned elements that are present at the current state of the metamodel being developed, and the available code generating feature, are presented in section 5 of this paper. It should be noted that agent definitions, as in listing 2, define an agent class, rather than an individual agent. This is required by the way of running an agent, i.e. creating a working instance of a SPADE agent, shown in listing 3. The agent in listing 3 is named `Olorin`, is an instance of agent class `Wizard`, uses unique local name of `Olorin@127.0.0.1`, and its password is `friend`.

```

1 class NiceBehaviour(spade.Behaviour.OneShotBehaviour):
2     """Behaviour without an agent."""
3     def _process(self):
4         print "Agent is behaving nicely."

```

Listing 1. Essential code of a BehaviourName behaviour

```

1 class Wizard(spade.Agent.Agent):
2     """Definition of agent class Wizard"""
3     def _setup(self):
4         print "This Wizard is alive!"
5         b = NiceBehaviour()
6         self.addBehaviour(b, None)

```

Listing 2. Essential code of an AgentName agent with added behaviour BehaviourName

```

1 if __name__ == "__main__":
2     a = Wizard("Olorin@127.0.0.1", "friend")
3     a.start()

```

Listing 3. Agent class is instantiated and an individual agent is created

## 5. Conceptual Example

The example showcased in this section is based on the idea of recipeWorld originally described by Fontana and Terna (Fontana and Terna, 2015). An idea of SPADE implementation of the recipeWorld elements is described in (Okreša Đurić *et al.*, 2016), along with some possible applications. In short, recipeWorld is an agent-based model that *simulates the emergence of a network out of a decentralized autonomous interaction*. Combination of agent-based modelling and network analysis is argued to *increase enormously the potential of complexity-based policies*. Principal elements of the recipeWorld are as follows:

- recipes - prerequisites for achieving a certain end, usually viewed as steps that vary in number;
- orders - specific ends to be pursued, concretised as objects containing technical information and basic data about order instances;
- agents - problem-solving cores that are either active or inactive, and are able to service some of the steps of a recipe.

Recipes are basically immutable series of codes representing various order components that demand certain services in order to be fulfilled. For example, the following is a representation of a recipe: [3 2 5 1 4] that demands access to the services as follows: first, service 3 has to be performed, followed by service coded 2, and so on until the element that needs access to service number 4. Once all the elements of an order have been dealt with by their corresponding services offered by other elements of the system, the order is finished, i.e. the pursued end is met. Figure 1 shows basic elements of the recipeWorld model: two orders (green) named by their ID numbers (7107 and 8714), with each having a single recipe of length two ([1 2] and [2 1] respectively), and two servicing agents (Factory59498 and Factory57780) serving up to two services each. Although simple in definition, recipeWorld can be adjusted and applied to various application domains, two of which are described in (Okreša Đurić *et al.*, 2016).

In terms of the metamodel being developed, several elements can be recognised, as is described below, and shown in Fig. 2.

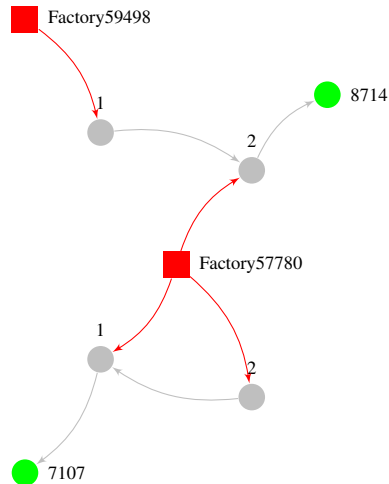


Figure 1. Principal recipeWorld elements: orders (green), recipe elements (gray), and service agents (red)

**Organisation Unit** Due to the active idea of roles containing most of the definitions specifying an agent, including their behaviour, knowledge, and interaction with other agents and their environment, only one class of agents is defined as an organisation unit. Instances of this agent class are further designated to play a certain role. Upon initiation of role-playing, the agents initialise all the necessary elements that identify them as individuals playing the given role (e.g. appropriate behaviours). Each individual agent has some individual knowledge reflecting their currently known situation in the system, e.g. what are the recipe parts of an order, and what is the status of each of them.

**Roles** Conforming to the description of essential elements of the recipeWorld, two roles are identified. **Factory** and **Order** roles can be played by *tabula rasa* individuals. Each of the proposed roles can access a common domain ontology containing all the needed knowledge of the system. Furthermore, each of the modelled roles has a number of behaviours at their disposal. E.g. an individual agent playing the **Factory** role will learn the following behaviours: **AnswerQuery**, **Produce**, and **Initialise**, which, combined, allow the agent to fulfill certain goals.

**Processes** Each of the modelled roles has specified behaviours specialised for certain outcomes. These behaviours are grouped in processes that can be used to achieve a certain goal. Goals achievable by a specific process of a given role are shown in the model. Processes, according to their respective goal-achieving features, are one of the essential elements for finding an appropriate role to be played by a certain individual at a certain point in time.

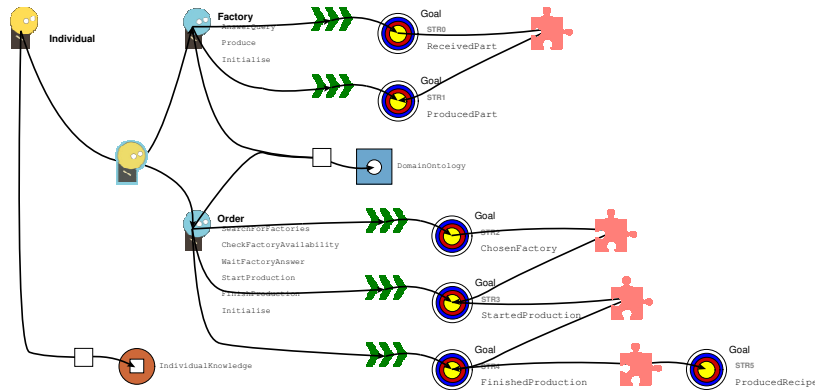


Figure 2. *recipeWorld* modelled using the current version of the metamodel being developed

**Goals** Several goals are identified based on the recognised states of the system, and of every individual agent. A prominent feature of goals is their hierarchy, i.e. relation of goals and subgoals. For example, goal `ProducedRecipe` is the ultimate goal of role `Order`, and it has a subgoal `FinishedProduction`. Individuals can be introduced to goals in various ways, which are not pondered in this paper.

**Knowledge** Two knowledge artefacts are shown in Fig. 2. One represents individual knowledge of an individual agent, e.g. what are the recipe parts of an order, and what is the status of each of them. The other knowledge artefact allows an agent playing either of the available roles to access an ontology describing the system as a whole.

When code generating feature of the metamodel is used, essential SPADE skeleton is generated. Organisation unit is generated as an agent class, behaviours of the modelled roles are defined as well, and process-goal pairs are turned into knowledge accessible to agents. SPADE agent class code generated from the metamodel is shown in listing 4.

```

1  import spade
   from RoleBehaviours import *
3
   class OrgUnit43Agent(spade.Agent.Agent):
4       '''Bear skeleton for agent type OrgUnit43Agent'''
5
6       class ChangeRole(spade.Behaviour.OneShotBehaviour):
7           '''Behaviour ChangeRole of OrgUnit Agent'''
8           def _process(self):
9               print 'Agent: behaving ChangeRole'
10
11          def _setup(self):
12              print 'OrgUnit43Agent: running'
13              cR = ChangeRole()
14              self.addBehaviour(cR, None)
15

```

Listing 4. SPADE agent class from the model shown in Fig. 2

## 6. Discussion

The example shown in the previous section may not be a typical example to be modelled using the metamodel being developed, but it does give a nice overview of the features currently included in the said metamodel. Notwithstanding, seemingly simple construction and only a few of the essential elements of the system do not prove the example to be trivial. Indeed, the *recipeWorld* can be run with large numbers of agents, thus creating a



complex environment that could simulate successful planning, decision making, and negotiation techniques. Two examples of possible applications of recipeWorld are described in (Okreša Đurić *et al.*, 2016).

The layers that are visually distinguishable in Fig. 2, creating a number of almost aligned columns, are approximately the layers that are to be developed for the final version of the metamodel. It is clearly visible from the example model that even with only a couple of concepts to be modelled, a complex layout and a potentially untidy overview are almost unavoidable. Therefore, it may prove useful to move groups of concepts to distinct theme-related layers, e.g. one layer that would emphasise hierarchical structure of the modelled goals, another layer containing only relations of various roles, etc.

No organisation-level modelling is shown in the example in this paper, since organisations are not innately a part of the recipeWorld model. The concept of organisations could be included in the model though, thus permitting modelling of e.g. more complex orders using simple order agents. Furthermore, some further development is needed for effective modelling of organisations using this metamodel.

Everything stated thus far in this section describes some indicators of the current state of the metamodel being developed. Although a lot of work remains to be done, some ground results are shown in this paper. Most notable elements that still have to be implemented are: organisation dynamics modelling, visualised by using graph grammars and described using e.g. temporal logics; and layered modelling, described earlier in this paper. Mechanisms of various concepts, and their relations, will have to be refined as well. An important envisioned feature of the final version of the metamodel being developed is generating programming code in multiple programming languages: during this research, the examples will mostly feature SPADE concepts, but the ultimate goal of the metamodel being developed is to provide mechanisms for generating MAS elements for various MAS implementations, SPADE being only one of them.

In addition to the elements stated above, future development of the ontology will have an impact on features of the metamodel. Since the ontology will feature chosen organisational concepts applicable to LSMAS, and the metamodel being developed is in tight connection with the ontology being developed, changes introduced to one of these two elements will affect features of the other, as proposed earlier in this paper, in chapter 3.

## 7. Conclusion

The proposed metamodel being developed is inherently aimed at modelling LSMAS enriched with organisational features. Elements enabling organisational modelling of LSMAS present quite a novel addition to LSMAS modelling. The most notable novelty envisioned in the final version of the mentioned metamodel is modelling organisation dynamics. According to what is mentioned earlier in this paper, organisation dynamics represents a feature most welcome to modern applications of LSMAS. Since one of the prominent application domains of LSMAS is the Internet of Everything, it is apt to consider modelling dynamics of various kinds of organisations formed in turbulent environments of decentralised systems of intelligent artificial agents. Furthermore, considering the short discussion on ACMAS vs. OCMAS presented in chapter 1 of this paper, it may be concluded that organisational features are a welcome concept indeed in large-scale complex systems comprising artificial intelligent agents.

One of the main features of MAS in general, and LSMAS by specification, is distributed, non-centralised, mechanics. Such a feature supports the idea of applying the metamodel being developed to various application domains of LSMAS.

Specification of the idea of organisation dynamics will be covered in more detail as a part of the future research, along with improvements of the current state of both the ontology and the metamodel. Further examples shall be covered with newly developed features of the mentioned metamodel as well.

## 8. Acknowledgements

This work has been supported in full by the Croatian Science Foundation under the project number 8537. My supervisor is to be acknowledged here as well, for generously helping me during my research.

## 9. References

- Abbas, H. A., Shaheen, S. I., and Amin, M. H., 2015. Organization of Multi-Agent Systems: An Overview. *International Journal of Intelligent Information Systems*, 4(3): 46–57. ISSN 2328-7675. doi: 10.11648/j.ijiiis.20150403.11.
- Barella, A., Ricci, A., Boissier, O., and Carrascosa, C., 2012. MAM5: multi-agent model for intelligent virtual environments. In *10th european workshop on multi-agent systems (EUMAS 2012)*, pp. 16–30.
- Corkill, D. D. and Lander, S. E., 1998. Diversity in agent organizations. *Object Magazine*, 8(4): 41–47.
- De Wolf, T., 2004. Emergence and Self-Organisation: a statement of similarities and differences. *Proc. of the 2nd Int. Workshop on Engineering Self*, pp. 96–110.
- Dignum, V., 2009. The Role of Organization in Agent Systems. In Dignum, V., editor, *Handbook of Research on Multi-Agent Systems*, pp. 1–16. IGI Global. ISBN 9781605662565. doi: 10.4018/978-1-60566-256-5.ch001.
- Fernández-López, M., Gómez-Pérez, A., and Juristo, N., 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *AAAI-97 Spring Symposium Series*, SS-97-06: 33–40. doi: 10.1109/AXMEDIS.2007.19.
- Fontana, M. and Terna, P., 2015. From Agent-based models to network analysis (and return): the policy-making perspective. *Working Paper Series*, 07.
- Gregori, M. E., Cámara, J. P., and Bada, G. A., 2006. A jabber-based multi-agent system platform. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems - AAMAS '06*, page 1282. ACM Press, New York, New York, USA. ISBN 1595933034. doi: 10.1145/1160633.1160866.
- Iqbal, R., Murad, M. A. A., Mustapha, A., and Sharef, N. M., 2013. An analysis of ontology engineering methodologies: A literature review. *Research Journal of Applied Sciences, Engineering and Technology*, 6(16): 2993–3000. ISSN 20407459.
- de Lara, J. and Vangheluwe, H., 2002. Using AToM as a Meta-CASE Tool. In *ICEIS*, volume 2, pp. 642–649. Lofgren, E. T. and Fefferman, N. H., 2007. The untapped potential of virtual game worlds to shed light on real world epidemics. *Lancet Infectious Diseases*, 7(September): 625–629. ISSN 14733099. doi: 10.1016/S1473-3099(07)70212-8.
- Maliković, M. and Schatten, M., 2015. Artificial Intelligent Player's Planning in Massively Multi-Player On-Line Role-Playing Games. In Hunjak, T., Kirinić, V., and Konecki, M., editors, *Central European Conference on Information and Intelligent Systems 2015*, pp. 3–10. Faculty of Organization and Informatics, Varaždin, HR.
- Okreša Đurić, B., 2016. Organizational Metamodel for Large-Scale Multi-Agent Systems. In de la Prieta, F., Escalona, M. J., Corchuelo, R., Mathieu, P., Vale, Z., Campbell, A. T., Rossi, S., Adam, E., Jiménez-López, M. D., Navarro, E. M., and Moreno, M. N., editors, *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*, Advances in Intelligent Systems and Computing 473, chapter 8, pp. 387–390. Springer International Publishing, Seville, ES. ISBN 978-3-319-40158-4. doi: 10.1007/978-3-319-40159-1\_36.
- Okreša Đurić, B., Tomičić, I., and Schatten, M., 2016. Towards Agent-Based Simulation of Emerging and Large-Scale Social Networks. Examples of the Migrant Crisis and MMORPGs. *European Quarterly of Political Attitudes and Mentalities EQPAM*, 5(4): 1–19.
- Schatten, M., 2014. Organizational Architectures for Large-Scale Multi-Agent Systems' Development: An Initial Ontology. *Advances in Intelligent Systems and Computing*, 290: 261–268. doi: 10.1007/978-3-319-07593-8\_31.
- Schatten, M., Grd, P., Konecki, M., and Kudelić, R., 2014. Towards a Formal Conceptualization of Organizational Design Techniques for Large Scale Multi Agent Systems. *Procedia Technology*, 15: 577–586. ISSN 22120173. doi: 10.1016/j.protcy.2014.09.018.
- Schatten, M. and Okreša Đurić, B., 2015. A Social Network Analysis of a Massively Multi-Player On-Line Role Playing Game. In Kang, B., editor, *Proceedings of the 4th International Conference on Modeling and Simulation*, pp. 37–42. IEEE, Jeju Island, Korea. ISBN 9781467398282. doi: 10.1109/MAS.2015.19.

- Schatten, M. and Okreša Đurić, B., 2016. Social Networks in «The Mana World» - an Analysis of Social Ties in an Open Source MMORPG. *International Journal of Multimedia and Ubiquitous Engineering*, 11(3): 257–272. doi: 10.14257/ijmue.2016.11.3.25.
- Schatten, M., Ševa, J., and Tomičić, I., 2016. A roadmap for scalable agent organizations in the Internet of Everything. *Journal of Systems and Software*, 115: 31–41. ISSN 01641212. doi: 10.1016/j.jss.2016.01.022.
- Schatten, M., Tomičić, I., and Okreša Đurić, B., 2015. Multi-agent modeling methods for massively Multi-Player On-Line Role-Playing Games. In Biljanović, P., editor, *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1256–1261. IEEE, Opatija, HR. ISBN 978-9-5323-3082-3. doi: 10.1109/MIPRO.2015.7160468.
- Schreiber, G., Raimond, Y., Manola, F., Miller, E., and McBride, B., 2014. RDF 1.1 Primer. <http://www.w3.org/TR/rdf11-primer/>. Accessed: 2 Dec 2016.
- Van Dyke Parunak, H. and Odell, J., 2001. Representing social structures in UML. In *Proceedings of the fifth international conference on Autonomous agents - AGENTS '01*, pp. 100–101. ACM Press, New York, New York, USA. ISBN 158113326X. ISSN 16113349. doi: [10.1145/375735.376008](https://doi.org/10.1145/375735.376008).
- W3C OWL Working Group, 2012. OWL 2 Web Ontology Language Document Overview (Second Edition). <http://www.w3.org/TR/owl2-overview/>. Accessed: 2 Dec 2016.
- Žugaj, M., 2007. *Znanstvena istraživanja u društvenim znanostima i nastanak znanstvenog djela*. Tonimir, Varaždinske Toplice, CRO.

