Applying the MAIA Platform to Model a Manure-based Energy System in the Netherlands

Amineh Ghorbani, Huib Aldewereld, Virginia Dignum, Gerard Dijkema
Faculty of Technology, Policy and Management, Delft University of Technology, a.ghorbani@tudelft.nl

Abstract. Implementing social concepts such as roles, norms and culture in agent-based models is complicated, especially if the modeller is not an experienced programmer. In this paper we present a web-based application that guides the modeller in defining social concepts in a way that would be feasible to code in an agent-based model. We illustrate the functioning of the software through a case study - manure-based bio-gas energy system. Not only does this software provide the means of conceptualizing an agent-based model in a formal manner, it also facilitates automatic generation of code.

Keywords: agent-based modeling, conceptualization, meta-model, simulation platforms

1 Introduction

Agent-based modelling (ABM) is a powerful tool for the study of social phenomena, especially if it is possible to implement social concepts besides individuals and their interaction to represent more of reality. To help identify and implement social concepts (e.g norm, institution, culture, role), some scientists advocate the use of conceptual frameworks (i.e. meta-models) [4, 3]. Although these frameworks facilitate the implementation of more complicated models, they are still far from mainstream. We see two explanations for this.

First, the set of concepts provided by the meta-models (e.g. [3]) are still not rich enough to cover the necessary concepts (e.g. institution, culture) existing in most social systems. Second, for those meta-models that are rich enough to model complex concepts (e.g [6]), the language is very close to programming code and therefore impractical for social scientists who are less familiar with programming.

In this paper, we show how the MAIA framework (Modelling Agent systems based on Institutional Analysis)[1] tries to overcome these two issues. Since MAIA is built on the Institutional Analysis and Development (IAD) framework [5] and other social theories ([2, 7]), it covers many social concepts not present in other modelling languages for social simulation.

In addition, in order to support the generation of computer-based simulations based on the MAIA framework, we follow a Model-Driven Engineering (MDE) approach. MDE approaches have in fact been advocated when constructing agent-based models of social systems [4]. With the web-based application developed for MAIA, the modeller can conceptualize a model without having to go into the details of programming. The outcomes of this conceptualization can be used by a programmer (who can be a different person) to implement the code without having expert knowledge. Furthermore, it is also possible to generate code from the XML output of model conceptualization.

To illustrate the diversity of social concepts in MAIA and also the functioning of the MAIA software, we conceptualize and implement the manure-based bio-gas energy system within rural areas in the Netherlands. The purpose of this case study is to identify those factors that influence the development of such system.

2 Background

Explicit model conceptualization, which entails describing the set of concepts that will constitute the “building blocks” of the model, is generally recognized to be a crucial step in building software models. Conceptualization leads modellers to better capture, analyse and understand what they are actually modelling [8]. In the social sciences, the institutional analysis and development framework (IAD) proposed by Ostrom [5] has been used successfully for many years for the conceptualization and analysis of complex social systems. In order to support systematic design of agent-based simulations for complex social systems, the MAIA framework extends and formalizes IAD. MAIA builds on the assumption that while understanding and explaining individual behaviour is extremely complex, social rules or institutions are more elicitable [7] and hence more readily identified and captured by modellers.
3 The Manure-based Energy System

Excess manure is a major problem for animal farmers in the Netherlands. Currently, farmers deal with the excess by distributing the manure on their own lands or neighbouring farms without animals. In a Manure-based Energy System (MES), animal farmers can produce bio-gas by processing manure which can then be fed to gas pipelines. Although, MES can be a source of income, farmers in the Netherlands do not have enough incentive to buy the technology. This is because the problem of excess manure is not solved with MES (no mass reduction after the processing of manure) and the subsidized technology is still expensive. The purpose of this case study is to identify the long terms prospects of using MES in order to create incentives for the farmers to invest in this technology.

4 The development process of an agent-based model using MAIA platform

In this section we briefly go through the steps that need to be taken in order to develop an agent-based model of the MES system. This case study was developed by two people. An analyst interviewed the domain experts, studied the literature, gathered information about the case and developed a conceptual model using the web-based application which we call the MAIA tool for short. A programmer used the output of the MAIA tool to implement a computer-based simulation.

4.1 Conceptualizing the MES

The web-based application of MAIA consists of 5 tabs: collective structure, constitutional structure, physical structure, operational structure and evaluative structure. The analyst who does not necessarily need a programmer fills in tables and diagrams in each of the five consequent tabs staring from the left shown in figure 1. The conceptualization continues until all the tabs have been completed. There are relations between the different tables and diagrams. The software automatically fills in much of the information accordingly.

The Collective Structure Tab

In this tab, the analysts identifies the agents that are relevant to the system (farmer, government, fertilizerDealer, manureDistributor) and fills in the required information for each agent. Currently, the fields in the tables are either free text such as the name or the properties of the agent (e.g. capital, age), or complex inputs which need to be defined in some other table and used here (e.g. possible roles = land owner). When defining the possible roles of the farmer agent, we can add a new role to the role table while still being in the agent table, by clicking on create new role in the drop-down menu.

The Constitutional Structure Tab

This tab defines the constitutional aspects of the manure-based energy system. The role table presents the roles within the system and their necessary fields (e.g. Objective and Entry Condition). Each agent can take from one to many roles. For example, the farmer agent can take the role of a landOwner, a bioGasProducer and an animalFarmer which are defined in the role table in figure 1. The institution table captures the rules, norms or shared strategies that the role-enacting agents must comply with. Furthermore, the constitutional tab defines the dependency between roles which is based on the objectives of the roles and defines the basis for agent interaction.

Fig. 1: The role table being completed in the Constitutional Structure tab of the MAIA tool.
The Physical Structure Tab

This tab defines the physical components of the system (e.g., cattle, pig, manure, digestate), the physical relationship between these components if necessary (each pig or cattle produces manure) and the physical connection between the components which were not necessary for this particular case.

The Operational Structure Tab

Within this tab, the analyst defines the dynamics of the system. The atomic components of the operational tab are actions. Each action has an action body which is in fact filled in by either an intrinsic behaviour of agents (e.g., those behaviours that agents perform without any thinking or decision making), behaviour of physical components (e.g., cattle produce manure) or institutional capabilities of roles (e.g., buy fertilizer, invest in technology). Each action body is placed in an action which has pre- and post-conditions. All actions are placed in action situations in the order in which they must run in the simulation. The specification of which role which agent takes to perform each action is also specified in this tab.

4.2 Implementing the Conceptualized Model

Following the MDE approach, there are specific rules for the transformation of each defined MAIA concept into Java code. The main simulation class has a method called the ActionArena() where all the action situations are placed in the specified order. The ActionArena() is called in the time loop for each agent instance. During initialization(), which is yet another method within the simulation class, agents are populated with primary roles. These roles may change during the run as the status of the system changes. The agents perform actions based on the specified conditions within each action class (e.g. BuyFertilizer(farmerAgent)). Each agent also has a decision making process for each institutional capability that he performs where he decides whether to comply with a rule or receive a sanction.

5 Conclusion

In this paper we briefly explained how an agent-based model can be developed using the MAIA tool. The conceptual framework provides guidelines on which concepts may need to be considered for any given social system. The analyst uses research, interviews, data and his own creativity to fill in the information requested by the web-based application. The programmer uses the information and the transformation rules to translate each concept to code the simulation program. The independence of these two responsibilities proved to be highly useful, as the domain expert can conceptualize an agent-based model without having to learn any programming. The output of the MAIA tool is an XML file. Currently, we are taking our approach one step further, developing mediator code to take over the translation of concepts to code. This would allow for the (semi) automatic generation of code which would benefit scientists who are not familiar with programming nor have programmers at hand, but would like to have executable agent-based models.

References