



Amineh Ghorbani, Pieter Bots, Virginia Dignum and Gerard Dijkema (2013)

MAIA: a Framework for Developing Agent-Based Social Simulations

Journal of Artificial Societies and Social Simulation 16 (2) 9

<<http://jasss.soc.surrey.ac.uk/16/2/9.html>>

Received: 11-Jun-2012 Accepted: 24-Nov-2012 Published: 31-Mar-2013

Abstract

In this paper we introduce and motivate a conceptualization framework for agent-based social simulation, MAIA: Modelling Agent systems based on Institutional Analysis. The MAIA framework is based on Ostrom's Institutional Analysis and Development framework, and provides an extensive set of modelling concepts that is rich enough to capture a large range of complex social phenomena. Developing advanced agent-based models requires substantial experience and knowledge of software development knowledge and skills. MAIA has been developed to help modellers who are unfamiliar with software development to conceptualize and implement agent-based models. It provides the foundation for a conceptualization procedure that guides modellers to adequately capture, analyse, and understand the domain of application, and helps them report explicitly on the motivations behind modelling choices. A web-based application supports conceptualization with MAIA, and outputs an XML file which is used to generate Java code for an executable simulation.

Keywords:

Modelling Language, Model-Driven Engineering, Institutions, Social Simulation, Meta-Model

Introduction

- 1.1 With the advent of software platforms (e.g., Repast (North et al. 2006), Netlogo (Tisue 2004), Swarm (Minar 1996)) in recent years, agent-based modelling (ABM) has gained popularity among social scientists. Building agent-based models is challenging because the modeller needs to consider the heterogeneity of agents across a population, and identify patterns of system behaviour that emerge from these agents' interactions (Macal & North 2010). To aid model development, some researchers provide guidelines on how to build agent-based models (see, for example, Gilbert & Troitzsch 2005, and see Drogoul et al. 2003 and Heath et al. 2009 for critical reviews). The general steps include model conceptualization/design, implementation, validation/verification, and analysis of data. 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. Precise, unambiguous descriptions lead to better capture, analyse and understand the domain being modelled (Winograd et al. 1996). A formal description of this set of concepts that describe a model is called a *meta-model* (Bézivin 2005; Schmidt 2006; Atkinson & Kuhne 2003).
- 1.2 In computer science, meta-models are widely used. On the one hand, a meta-model provides the conceptual richness needed for modelling a wide range of complex systems; on the other hand, it has the formal rigour needed for automatic translation of high-level system descriptions to executable software. However, although some researchers advocate using meta-models when constructing agent-based models of social systems (Hassan et al. 2009; Janssen et al. 2008; Sansores & Pavón 2005), this practice is still far from mainstream.
- 1.3 Sansores & Pavón (2005), Garro & Russo (2010), Bousquet et al. (1998) and Hassan et al. (2009) use meta-models in the tools they propose for ABM. However, their meta-models do not include social structures such as norms and cultures (which are important aspects of a social system), and their agent concept lacks features such as personal values and preferences that affect their behaviour. The meta-model proposed by Iba et al. (2004) only defines basic ABM concepts (e.g., agent, relation, entity). The meta-models proposed in agent-oriented software development literature (e.g., Adelfe (Picard & Gleizes 2004) and PASSI (Cossentino 2005)) are so close to programming language and so computationally complex that they fail to reach the broader community of social scientists. Developers of software platforms for ABM, such as Repast, Netlogo and Ascape, have recognized the necessity of meta-models for their platforms, and are beginning to reverse engineer meta-models that would fit these platforms (Janssen et al. 2008).
- 1.4 In social systems, sharing of resources and interactions take place under institutional structures. To understand and analyse such systems, we propose the meta-model MAIA^[1] (Modelling Agent systems based on Institutional Analysis) that remedies the aforementioned shortcomings. MAIA builds on the assumption that, while understanding and explaining individual behaviour is extremely complex, social rules or institutions are more elicitable (Scharpf 1997) and hence more readily identified and captured by modellers. Therefore, with this meta-model, we aim to describe those systems where individuals and institutions are the key components. To reach this goal, MAIA extends and formalizes the components of the Institutional Analysis and Development framework (IAD) (Ostrom 2005) that has been used successfully for many years in analysing social systems with institutional settings.
- 1.5 To support the use of the MAIA meta-model, we also propose (1) a web-based application that supports the conceptualization process in MAIA, and (2) a methodology that provides guidelines on how to produce executable code from a conceptual model.
- 1.6 The remainder of this paper is structured as follows. In Section 2 we motivate the use of the IAD framework as basis for our work. We will introduce the MAIA meta-model in Section 3, and explain the methodological aspects of MAIA in Section 4. In the final section, we will discuss the contributions and ultimate potential of MAIA, and make some concluding remarks.

Conceptualizing Social Systems

- 2.1 To explain the key components of a social system, in this section, we use the IAD as the general framework. We describe the concepts and relations in this framework with other relevant theories in the social sciences.
- 2.2 Individuals are the key steering components of a social system (Hedström & Swedberg 1996). The behaviour of individuals depends on their demographic properties, preferences, personal values, and capabilities, as well as their physical and social context. This social context not only comprises the behaviour of other individuals; it is also structured by the interactions between individuals (Giddens & Turner 1988), which in turn are formed by the institutional settings (Giddens 1984; Scharpf 1997, p. 12).
- 2.3 Ostrom (1991) defines an institution as "the set of rules actually used by a set of (actors) to organize repetitive activities that produce outcomes affecting those (actors) and potentially affecting others". In other words, institutions are major components of a social system which affect the way actors behave (Scharpf 1997, p. 21; Giddens & Turner 1988).
- 2.4 Although institutions set the necessary preconditions for individual interactions (Scharpf 1997, p. 21), individuals are not merely rule followers. They are intelligent and intentional entities who, depending on their individual characteristics, may decide to disregard institutions under certain conditions.

Analyzing Systems from an Institutional Perspective

- 2.5 Ostrom et al. (1994) propose the IAD as an institution-driven tool for (1) understanding the underlying structures of a social system (the left-hand side of Figure 1), (2) capturing the operational environment (the centre of Figure 1), and (3) observing the patterns of interaction and outcomes, given a set of evaluation criteria (the right side of Figure 1). The result of this social system analysis is used to give feedback to the system, with the intention to support institutional change (the feedback loops in Figure 1).
- 2.6 In the IAD framework, institutions which are embedded in the social community are not the only factors that affect the operational context of a system. The physical world in which the community is situated, and the resources used or produced, likewise influences actor behaviour. The IAD framework views the dynamic context that comprises and is formed by (inter)acting actors as a set of action situations. In each action situation, actors interact: they communicate, exchange goods and services, and negotiate. An action situation consists of roles (or positions), actors (or participants), the actions actors perform, information related to the situation, expected outcomes of the situation, and costs and benefits. What happens in the operational context of a social system leads to patterns of interaction and outcomes that are judged on the basis of evaluative criteria defined by the analyst.

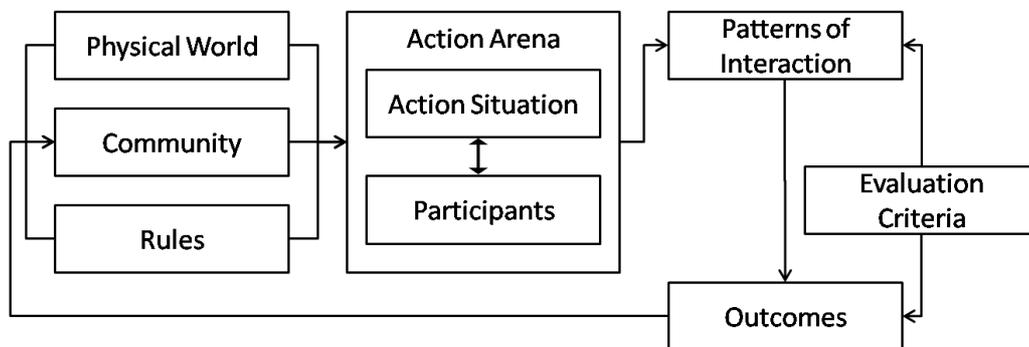


Figure 1. The IAD framework (Ostrom et al. 1994)

- 2.7 The IAD framework illustrated in Figure 1 has been in development for more than 30 years, and its concepts have proved to be robust in numerous case studies (e.g., Yandle & Dewees 2003; Gordillo & Andersson 2004; Wynne 1989; Oakerson 1992). Several case-specific agent-based models that have been developed with the IAD perspective (e.g., ABM of Land change (Manson 2005), ABM for Natural resource management (Bousquet et al. 1998), common pool ABM experiments (Deadman et al. 2000)) confirm its potential for agent-based model development.
- 2.8 The concepts in the IAD framework are easy to grasp by analysts, and facilitate stakeholder interaction; however, they are not sufficiently unambiguous to be able to formulate a computer simulation. In order to use this framework as a basis for a conceptualization framework for agent-based social simulation, we need to:
1. Formalize actors and their internal characteristics/attributes independently from the roles these actors take and the rules they follow. This would provide more insight into individual behaviour and decision making. We will use other social theories and frameworks for this purpose (Scharpf 1997; Giddens & Turner 1988; Hedström & Swedberg 1996).
 2. Clarify the meaning of some IAD concepts and additional system components by defining their characteristic attributes² in more detail.
 3. Redefine the usage of the right-hand side components (in Figure 1) of the IAD: patterns of interactions, outcomes, and evaluation criteria. Where in the IAD these components are the outcomes of the operational environment of the actual social system, we must now define and formalize them as outputs of a software model.
 4. Formalize the relationship between the components of the IAD to ground the behaviour of the system and clarify the relation between various components of the software model.
- 2.9 Furthermore, to make the framework truly practical for developing an agent-based computer simulation, we need to clarify the process of mapping the IAD concepts to programming elements. For example, some IAD concepts may be represented as classes in object-oriented programming while others would

be implemented as methods. Guided by discussion with IAD experts and experiences in four case studies (Steubing et al. 2011; Ligtvoet et al. 2011; Ghorbani et al. 2011; De Korte 2012), we have extended the IAD into a meta-model for conceptualizing systems for agent-based modelling.



Modelling Agent Systems based on Institutional Analysis

- 3.1 In this section, we present the MAIA meta-model. A meta-model offers the vocabulary needed to describe and reason about a model. That is, a meta-model makes statements about what can be expressed in a (valid) model. As discussed in the previous section, IAD concepts are a suitable vocabulary to describe and analyse collective behaviour, i.e. the structures and mechanisms of social order and cooperation governing the behaviour of several individuals.
- 3.2 Following the IAD framework, we organize the MAIA meta-model into five structures that serve as place holders for related concepts:
1. Collective Structure: actors (referred to as participants in the IAD) and their attributes.
 2. Constitutional Structure: the social context.
 3. Physical Structure: the physical aspects of the system.
 4. Operational Structure: the dynamics of the system.
 5. Evaluative Structure: the concepts that are used to validate and measure the outcomes of the system.
- 3.3 During the conceptualization process, these structures are gradually filled by the modeller with the details of the phenomena being modelled. In the following description of MAIA, the concepts are written in *italic* font when first introduced, and case-specific concepts are in `typewriter` font. A complete overview of the MAIA concepts in these five structure and their relations is presented as a UML class diagram in Appendix 1.

Working scenario

- 3.4 We present the concepts and relations that constitute the MAIA meta-model^[3] with the help of an illustrative example taken from one of the four case studies in which we have applied MAIA: informal backyard recycling in Bangalore, India (Sheoratan 2011).
- 3.5 Informal backyard recyclers in India handle e-waste in an unskilled, harmful and inefficient way in order to extract valuable materials (Ha et al. 2009). These activities take place in a social context with many unwritten and sometimes imperceptible rules, norms, and shared strategies among the agents. Child employment and unsafe extraction (causing health and environmental hazards) are the major growing problems of this sector.
- 3.6 As a solution, the Indian government wants to introduce professional recycling companies that would take over the precious metal extraction. These companies would take the dismantled parts from the backyard recyclers, extract gold, and return the cash value of this gold to the backyard recyclers. Data show that the revenue received from professional companies is higher than the value of the inefficiently extracted gold. However, when backyard recyclers employ children and use unsafe chemicals, their revenue from this practice is higher than what they would receive from professional companies. This may explain why the government policy has not been successful.
- 3.7 One of the major questions that we aimed to answer by means of agent-based simulation was how the rules for fining recyclers for child labour and harmful extraction would influence the economic situation of the recycling units and increase their incentive to work with professional recycling companies. In the following sections, we use a simplified version of this case study to explain how an agent-based model of a social system can be developed using MAIA.

Agents and the Collective and Constitutional Aspects

- 3.8 The e-waste recycling system is viewed as a social system with several types of actors which we call *agents* in MAIA. Agents can represent individual as well as composite actors (Scharpf 1997, p. 43). A composite agent can represent a collection of agents such as a company or a family, and its constituents may in turn be composite agents. In our model, the agent types `worker` and `government representative` represent individual actors, while a `dealer` in old computers or a `professional recycling company` are composite agents.
- 3.9 Independent of the roles they may assume in the society, agents have *properties, personal values, belongings, information, and intrinsic behaviours*. In our model, the relevant properties of a `worker` are `age` (adult or child), `skill level`, `experience`, `money` and the level of `risk` he is willing to take in his job. A `government representative` has `money` and may be `corrupt` (a Boolean property). The `worker` agents have `wealth` and `safety` as their personal values^[4]. Agents may own physical components and information. For example, all workers have `tools` and know the `price of gold` and the `price of old computers`. Agents may have intrinsic behaviours, irrespective of the role they are taking in the society. For example, all workers `lose energy` when they work.
- 3.10 Agents make *decisions* about the tasks they perform. We assume that each decision making behaviour^[5] requires a criterion. For example, to model that more risk-taking bosses are more inclined to employ children when hiring employees, a decision criterion could be:
- $$\text{decisionCriterion} = \text{worker.skillLevel} * (\text{worker.isAdult} * (1 - 2 * \text{boss.riskLevel}) + \text{boss.riskLevel})$$
- In this way, bosses will prefer (i.e., have a higher value for the decision criterion when considering) highly skilled workers, and will progressively consider child labour as their risk level rises above 0.5.
- 3.11 In sum, the Collective Structure specifies the attributes of all agents in the model (Figure 2).

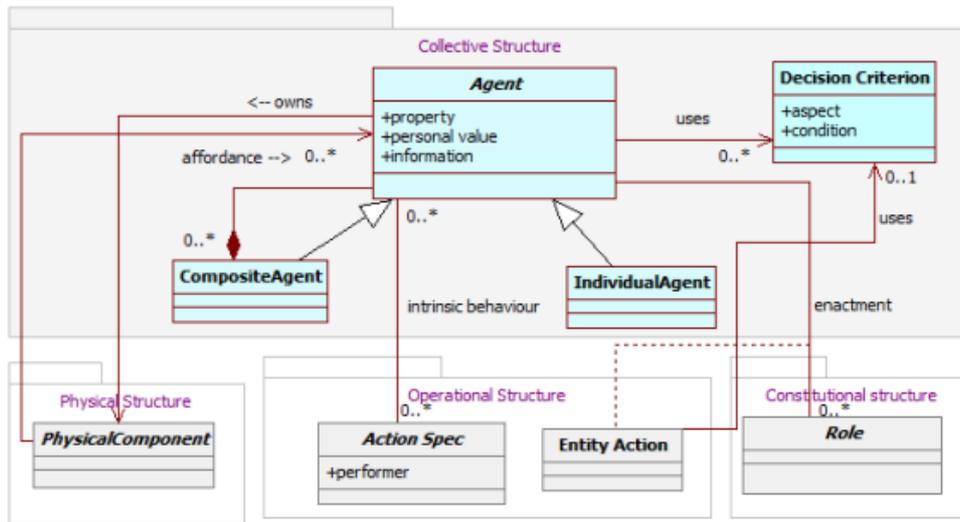


Figure 2. The Collective Structure in MAIA

- 3.12 To take part in the society, agents enact *roles* (referred to as positions in the IAD). A role is an abstract representation of a set of activities that are performed according to some rules in order to reach social objectives (Dignum 2004; Ferber et al. 2004). A *worker* can take the role of a *segregator* who dismantles computers, a *refurbisher* who refurbishes computer parts, or an *extractor* extracting gold. He may also take the role of a *unit boss* and hire other workers.
- 3.13 Actors may take a role in the society, only if an *entry condition* is met. To become a unit boss, *sufficient money* is required. To become a segregator, refurbisher, or extractor, the agent needs to *have tools*. An agent can take multiple roles in a model, and the same role can be assumed by multiple agents, simultaneously or sequentially. This is specified through the entry condition. For example, a *worker* having the role of *segregator*, *refurbisher* and *extractor*, can also take the role of a *unit boss* at the same time, provided that the conditions are met.
- 3.14 An *objective* is the expected result of a role (Dignum 2004). Segregators, refurbishers and extractors have increase of *income* as their objective. The objective of a *rule enforcer* is reduction of hazards and *child labour*. Role dependency forms the basis of relationship between agents. Roles depend on other roles to achieve their objectives. The *segregator*, *refurbisher* and *extractor* agents depend on their *boss* for income. The *rule enforcer* depends on *unit bosses* for reducing hazards and *child labour*. This objective dependency reflects the idea that it is the institutional setting which initiates relationships between agents (Scharpf 1997).
- 3.15 When agents take roles, certain *capabilities* (or responsibilities) become available to them. Segregators can *dismantle computers*, refurbishers can *make refurbished products*, extractors can *extract gold*, the *rule enforcer* can *fine unit bosses*, and the *dealer* can *sell computers* and *buy refurbished parts*.
- 3.16 A society with a diverse set of role-enacting agents functions only if there are rules and conventions that govern agent behaviour (Ostrom 1991). Such rules and conventions are institutional statements that can be formulated using the ADICO syntax (Crawford & Ostrom 1995). The acronym ADICO refers to the five elements that an institutional statement can comprise: Attributes (the designated roles), Deontic (prohibition, obligation, permission), aim, Condition (for the institution to hold), and 'Or else'.
- 3.17 The aim of a statement is an action taken by an agent defined as a capability of the role that the statement is part of (i.e., the attribute of the statement). The 'Or else' specifies the *unique and explicit* sanction that applies if an agent does not comply with an institution. The 'Or else' can itself be an institutional statement.
- 3.18 Crawford & Ostrom (1995) show that institutional statements can be categorized into three types: rules, norms and shared strategies. Table 1 shows some institutional statements and their types for the e-waste example. Statements containing all the five ADICO components are referred to as *rules*. In the first example in the table, the agent would get an *explicit unique* sanction from the government if he does not comply with this rule. When there is no explicit sanction (i.e., no 'Or else') the statement is referred to as a *norm*. In examples 2-4 in the table, there may be consequences for non-compliance, but they are neither unique nor clear. Therefore, these statements are considered to be norms. Finally, if there is no deontic flavour to the statement, that statement is called a *shared strategy*. In Statement 5 in the table, there is no obligation that the agents must pay half salary to children, nor is there a sanction for non-compliance; most bosses just happen to take the same strategy.

Table 1: Different types of institutional statements in the e-waste example.

Type of Statement	Statement
1 Rule	A unit boss may not hire workers if they are children or else he will be fined
2 Norm	A unit boss must pay minimum of \$50 per day to a refurbisher
3 Norm	A segregator must dismantle at least 100kg of computer per day
4 Norm	Segregators, refurbishers and extractors may not sell products if they are employees
5 Shared Strategy	Unit Bosses pay half salary if the employee is a child

- 3.19 Taking a role in the system does not force agents to follow the rules associated with that role. Based on the agent's properties and personal values among other conditions, the agent may decide not to follow an institutional rule, and give priorities to his personal values instead. In fact, for every action (i.e., the

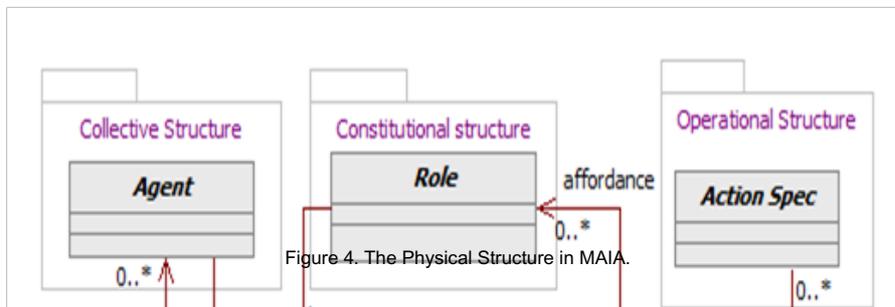


Figure 4. The Physical Structure in MAIA.

Agents and their Operational Environment

3.23 The Operational Structure (Figure 5) describes the dynamics of the agents' influence on the system state. It defines the actions that the entities (agents, roles and physical components) of the system are responsible for, and the (partial) order in which these actions are performed. In every time step in the simulation, each agent enters the action arena to explore the actions he may be able to execute. Each simulation has exactly one action arena that is defined by a list of *action situations*, where each action situation describes the order in which a number of related *entity actions* take place using *plan* specifications. For the informal e-waste recycling community, the computer recycling action arena consists of these four action situations:

- **Market:** Unit bosses (*precondition:* have segregators, have money) and segregators (*precondition:* have money) will buy old computers. Workers in any role will sell products (i.e., refurbished parts, gold and waste) (*precondition:* have product) on the market.
- **Production:** Segregators will dismantle computers, refurbishers will make refurbished parts, and extractors will extract gold. Extractors can decide to use unsafe chemicals for extraction, which will then decrease their energy.
- **Employment:** those agents who have sufficient money may form recycling units and become unit bosses. They will calculate hiring need, search for suitable employees, pay employees and fire employees.
- **Safety inspection:** the rule enforcer randomly inspects recycling units to find out whether they employ children or do dangerous extraction. The rule enforcer can fine unit bosses if they are not following the rules. If the rule enforcer is corrupt, he can be bribed and will then not fine the unit boss.

3.24 Each of the entity actions (e.g., buy old computers) listed in the action situations has a *precondition*, which tests the actual feasibility of performing one action (e.g., have money), and a *postcondition*, which specifies the update in the system state (e.g., increase in the number of computers, and decrease of their money). The agent may *enact a role* to perform an action. Furthermore, every entity action may be associated with a *decision making* process and an *institution* that the agent must take into consideration. For example, the agent who enacts the role of a unit boss must decide whether to employ a worker, and during this decision making he takes the child labour institution into account. He may decide to employ the worker, even if this is a child, based on his personal values and other factors that influence his decision.

3.25 The order of entity actions in an action situation is specified by a plan. Plans are defined recursively using these four types of plan:

1. *atomic plan:* the plan consists of a single entity action (e.g., become unit boss).
2. *sequence:* consists of a set of plans that will be executed in the specified order (e.g., select employee, hire employee).
3. *alternative:* consists of a set of plans from which one is selected randomly (with equal probability).
4. *loop:* consists of a plan that is repeated for as long as a condition holds (e.g., process old computers until there is none left).

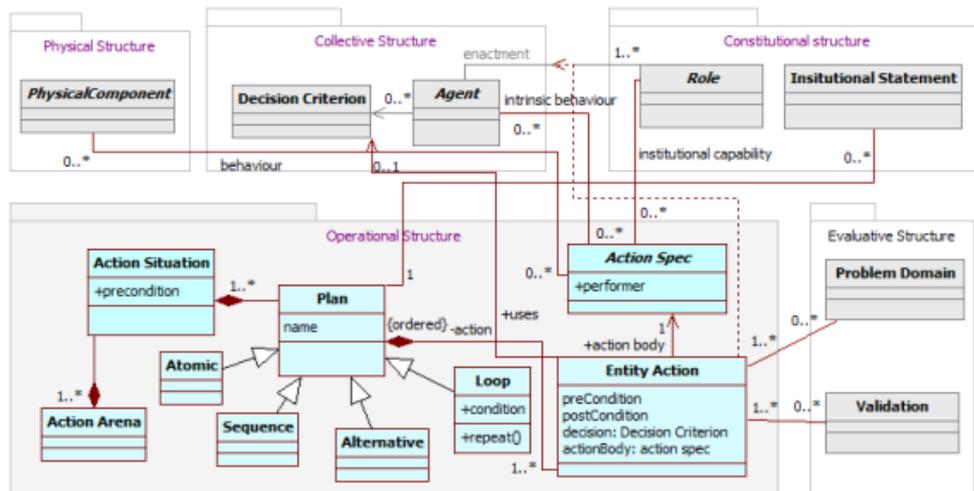


Figure 5. The Operational Structure in MAIA.

Answering questions using the agent-based model

3.26 The Evaluative Structure (Figure 6) is inspired by the right-hand side of the IAD framework (Ostrom et al. 1994) depicted in Figure 1. It should provide concepts with the help of which the modeller can indicate what patterns of interaction, evaluation, and outcomes she is interested in. In other words, the modeller should be able to identify those variables that can serve as indicators for model validity (is it sufficiently realistic?) and model usability (will its implementation help me to explore the question(s) I set out to address?).

3.27 To make sure that the model implementation does not violate real-world conditions, constraints can be specified for variables. In our e-waste example, we do not want to see negative volumes of old computers, a child worker cannot become a unit boss, and so on. With every validation variable, we associate the entity actions that actually influence its value. If there is a direct influence (e.g., the entity action sell computers decreases the number of computers), we set the *type* of this association to *direct*. If there is no direct influence (e.g., segregation turns computers to other products, which results

in segregators buying more computers, and therefore reducing their number) we set the type to `indirect`. Specifying the type of relation between an entity and a variable will help focus the analysis of the data after a simulation run.

- 3.28 To make sure that the model implementation will provide answers to questions about the system, the modeller can specify what variables are useful indicators for the problem domain. The issue we wanted to explore with the e-waste model was the distribution of income for a recycling unit during the simulation. A variable that can be used to give insight into this dynamics is the `recycling unit size`, because only those units that have high income can hire employees. To explain how the size is affected throughout the simulation, there is a direct relation between this variable and the `hiring` and `firing` entity actions. This variable also has an indirect relation with the `sell products` and `buy computers` actions. To explain this indirect relation, it would seem logical to look at the number of segregators in each recycling unit, because this is the variable that determines how many computers can be bought. Following the same line of reasoning, we define more outcome variables: `number of refurbishers per unit`, `number of extractors per unit` and `number of recycling unit` during the simulation run.
- 3.29 To facilitate the visualization of results, we define independent variables. For example, to draw a diagram for monitoring the `recycling unit size` during the simulation we define the independent variable `time step` for the problem domain variable `recycling unit size`.

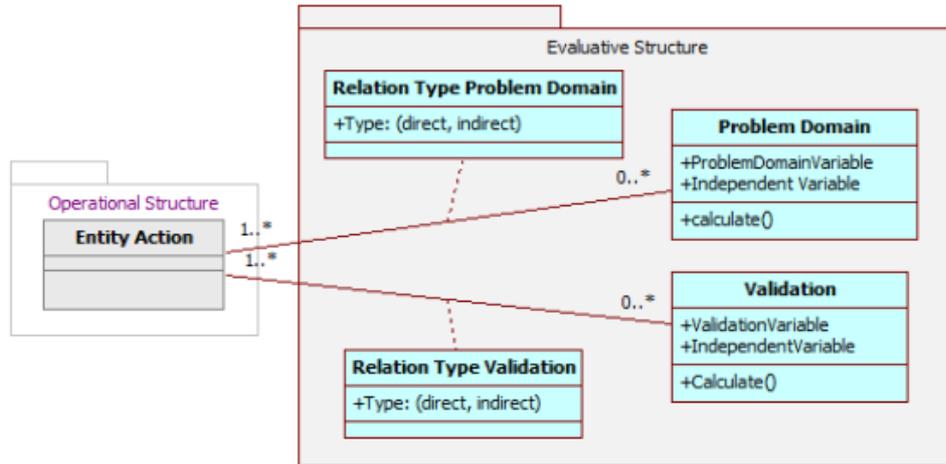


Figure 6. The Evaluative Structure in MAIA.

- 3.30 Together, the concepts defined in the five structures presented in this section constitute the MAIA meta-model for conceptualizing and implementing agent-based simulations of social systems. A comprehensive overview in the form of a UML class diagram can be found in Appendix 1.

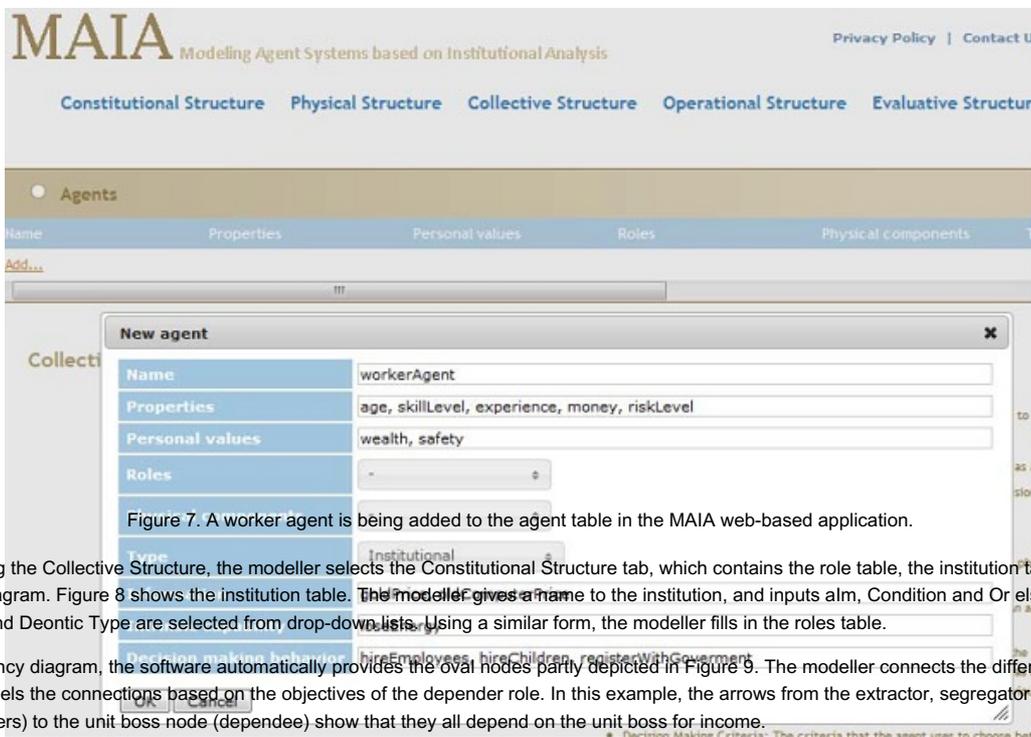


Methodological Aspects

- 4.1 We will now explain how the MAIA web-based application ('the MAIA tool') guides the conceptualization process of a social system in terms of the MAIA meta-model. We will then discuss how the MAIA tool facilitates model implementation by outlining how a MAIA-based conceptual model is translated into executable code.

The Conceptualization Process of a Social System

- 4.2 The conceptual model (MAIA-based model) of a social system is presented in eight tables, three diagrams, and two matrices which are completed by the modeller. The MAIA tool^[6] supports this process, and produces an XML file that can be transformed into computer code. This tool also saves data for documentation purposes.
- 4.3 The MAIA tool is organized in five tabs: one for each structure in the meta-model. Each tab contains tables and diagrams for the concepts of that structure.
- 4.4 Conceptualization starts by identifying the agents and their attributes, and completing the agent table in the Collective Structure tab. Figure 7 shows a worker agent being defined in the MAIA tool. An 'add' link in the role drop-down list provides the possibility to add new roles to the agent, while still continuing completion of the agent table. Besides physical components and possible roles, all the other attributes such as properties (e.g., `age`, `skillLevel`) and personal values (e.g., `wealth` and `safety`) are simple text inputs^[7]. Roles and physical components are selected from the definitions in associated tables.



- 4.5 After completing the Collective Structure, the modeller selects the Constitutional Structure tab, which contains the role table, the institution table and the role dependency diagram. Figure 8 shows the institution table. The modeller gives a name to the institution, and inputs alm, Condition and Or else as open text. The Attribute and Deontic Type are selected from drop-down lists. Using a similar form, the modeller fills in the roles table.
- 6.6 In the dependency diagram, the software automatically provides the oval nodes partly depicted in Figure 9. The modeller connects the different roles (nodes) and labels the connections based on the objectives of the dependor role. In this example, the arrows from the extractor, segregator and refurbisher nodes (dependers) to the unit boss node (dependee) show that they all depend on the unit boss for income.

Name	Attributes	Deontic type	Aim	Condition	Or else
hireRule	unitBoss	Prohibition	hireWorker	worker.age = child	fine
paymentNorm	unitBoss	Obligation	paySalary 50	per day	
productSellNorm	3 selected	Prohibition	sellProduct	employee = true	

Figure 8. The institution table for e-waste in MAIA software.

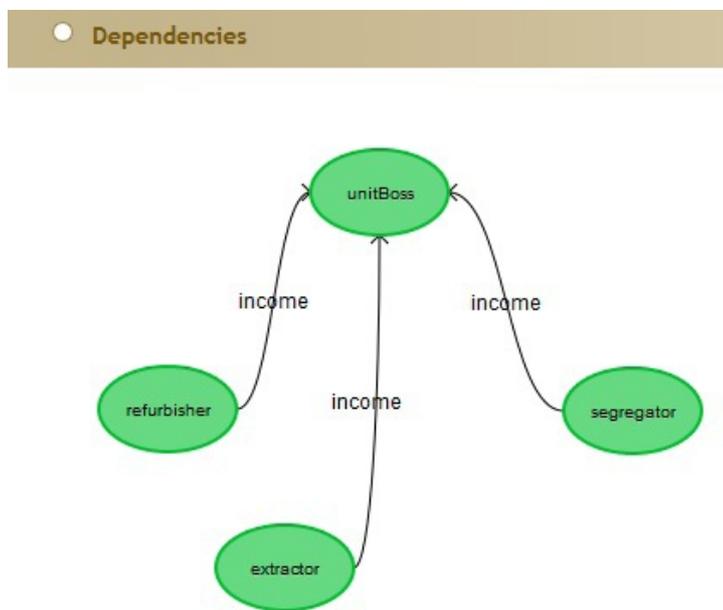


Figure 9. Objective dependencies between roles in the e-waste example.

- 4.7 The Physical Structure tab of the MAIA tool lets the modeller enter attributes in the physical component table, which is quite similar to the agent and role tables. Furthermore, it lets the modeller specify relations between components in a composition diagram and a connection diagram. The software generates the nodes (physical components) for these diagrams, while the modeller specifies the connections between these nodes (similar to the dependency diagram in Figure 9).
- 4.8 In the Operational Structure tab, the modeller makes use of the components defined in the other tabs to specify the dynamic structure of the model. Each entity action uses a capability or behaviour of previously defined entities (i.e., physical component, agent or role). Preconditions specify under which conditions an action occurs; postconditions are the ensuing changes in the system state. For example, to define the *fine employees* capability of the *rule enforcer* role, the modeller specifies the changes that occur when this event happens: *isFined* state of agent is set to true, the *money* for the agent in the role of a unit boss is decreased by some amount, and the *money* for the government agent is increased by the same amount. When an institutional capability is associated with an institution, the agent must make a decision whether to comply with that institution. Therefore, a decision criterion is always required.
- 4.9 As a convention, the modeller should use verbs for the names of actions (e.g., *hireEmployees*) and nouns for the names of action situations (e.g., *Employment*). The modeller orders the entity actions by defining plans, and placing them in action situations. An action arena table displays the action situations in order to show the general sequence of events taking place.
- 4.10 The modeller finally links the conceptual model to the expected outcomes of the agent-based model in the Evaluative Structure tab, which contains: the scope table^[6] and the validation table. In each row of the scope matrix, the modeller defines the problem domain variable, an entity action related to the variable, the type of relation between the problem domain variable and the entity action, and the independent variable. The validation table is similar to the scope table.
- 4.11 The information captured in the evaluative structure is not used for implementing the simulation, but rather for analysis of the results. It helps the modeller to specify the presumed influence of entity actions on the variables that will be giving measures of the outcomes. The independent variables will be used to draw charts.
- 4.12 During conceptualization, the modeller is continuously stepping backward and forward through the tabs to define more concepts or update the old ones. Once the modeller is satisfied with the conceptualization, and the procedure has been completed, she has the option to save or print the completed tables and diagrams, or export the data into an XML file that is used for code generation.

Model Implementation

- 4.13 The MAIA meta-model concepts can be used to produce code in different programming languages. Since MAIA concepts are organized into relational tables, they are especially straightforward to code using object-oriented programming languages. In the following, we describe the transformation process of a social system described in MAIA (a MAIA-based model) to an object-oriented simulation program.
- 4.14 The five structures in MAIA organize the concepts into different categories for storing the source files. Some of the general MAIA concepts (e.g., *Agent*, *PhysicalComponent*, and *Institution*) are abstract classes that domain classes (e.g., *WorkerAgent*) inherit from. Agents in a simulation are objects that take static descriptions of roles and check static descriptions of institutions to perform actions. The entry point for the agents to perform actions is the main simulation class.
- 4.15 Similar to other simulation platforms (e.g., Repast (North et al. 2006)), the main simulation class contains three tasks. First, it initializes the simulation by assigning default values and instantiating objects. Second, it gives the opportunity to each agent in each time step to execute an action. Finally, it analyses simulation data. Listing 1, shows the pseudo-code that a programmer or a translator software would produce for the main simulation class.

```

BEGIN Simulation E-waste
    Initialize_values
    FOR EACH timeStep IN (1..365)
        FOR EACH agent IN worker_agents, dealer_agents, government_agents
            ActionArena(agent)
        END FOR
    END FOR
    Analyze_data
END Simulation

```

Listing 1. The main simulation algorithm for the E-waste example.

- 4.16 In the e-waste example, each *timeStep* is equivalent to a day in reality. In each *timeStep*, all agents enter the *ActionArena* one by one. Listing 2 shows the *ActionArena* in the e-waste example with all the entity actions as procedures, and plans and action situations as comments. This is because in the current implementation, action situations and plans only specify the sequence of actions in the *ActionArena*. The 'lowest level' statements in Listing 2 (represented as procedures with an agent as the input parameter) denote atomic plans. Each agent can perform at most one action per *timeStep*. It does this by checking the entity actions one by one to see which one it can perform. As soon as an entity action is executed (whether successful or not) the agent exits from the action arena, keeping track of the plan it is performing, and the last completed step in that plan. When re-entering the *ActionArena* in the next time step, the agent returns to the next step of the plan, or the next plan.
- 4.17 In each *action situation* piece of code, plans show the order of entity action execution. For example in Listing 2, in the market situation, the agents performs a sequence of actions. In one time step they buy old computers and in the next time step they sell products. Note that interaction between agents also takes place within entity actions. When the agent enters an entity action, the other agents it might be interacting with, take a passive position. When those agents also enter the same entity action, the interaction may complete. For example, when the segregator wants to buy computers, he sets a price for the dealer. When the dealer enters the buy computers entity action, if he sees a segregator with a set price (a tuple), he may give the computer to the segregator and take the money. As another example of interaction, an agent finds a violator of a rule (e.g., child employment) and may fine him. It does this by going through the list of all agents and finding one that is violating the rule. Fining that agent implies that the violator agent's money is reduced.

```

BEGIN ActionArena(agent)

    //market action situation
    //plan type: sequence
    buy_old_Computers(agent)
    sell_products(agent)

    //production action situation
    //plan type: sequence
    dismantle_computers(agent)
    make_refurbished_parts(agent)
    extract_gold_safely(agent)
    extract_gold_unsafely(agent)

    //employment action situation
    //situation condition
    IF(agent IS unit_boss)
        //plan type: sequence
        form_recycling_unit(agent)
        calculate_hiring_need(agent)
        // plan type: loop
        WHILE(employee_not_found & workers > 0)
            findEmployee(agent)
        END WHILE
        //plan type: sequence
        pay_employees(agent)
        fire_employees(agent)
    END IF

    //safety inspection action situation

    //situation condition
    IF(agent IS government)
        //plan type: sequence
        find_unsafe_extraction(agent)
        find_child_employment(agent)
        //plan type: alternative
        fine_violators_bribed(agent)
        fine_violators_not_bribed(agent)
    END IF
END ActionArena

```

Listing 2. The action arena in the e-waste example.

- 4.18 In each *entity action*, the precondition checks whether the agent can perform the action and if so, the system state is updated accordingly (e.g., number of computers decreases). Agents execute actions in three different ways. In the first situation, if the body of the entity action is an intrinsic capability of an agent or behaviour of physical component, and preconditions hold, the agent executes the action. In the second situation, if the body is an institutional capability and there is a decision making process associated with the entity action, the agent *decides* to perform the entity action if preconditions hold. Finally, if there is also an associated institution and the body is an institutional capability, the agent decides *whether* to comply with the institution and to perform the action (when preconditions hold).
- 4.19 The purpose of this section was to give an idea of how MAIA concepts can be coded. Therefore, we only explained the overall implementation of the operational structure and did not go into the implementation details of the other structures. This will be the subject of our next publication.



Evaluation of the MAIA Meta-model

- 5.1 Our ambition in this research is to facilitate the development of more comprehensive agent-based models and bring ABM within the reach of more social scientists. To evaluate how close we have come to these objectives, we evaluate MAIA along four aspects: conceptual soundness and completeness reflect its potential for developing more comprehensive agent-based models; parsimony and usefulness show whether our modelling approach brings ABM within the reach of more scientists, especially those who are unfamiliar with programming.

Conceptual Soundness

- 5.2 We contend that grounding our meta-model in the IAD framework - which has matured through many different applications over three decades - adds to the conceptual soundness of MAIA. Through expert evaluation we verified that the added details and concepts (e.g., our detailing of the physical structure in terms of components, connections and composition) are in line with the ideas in the IAD. Furthermore, we made sure that there is a clear association between concepts in MAIA, and that all of the concepts and relations are used during model development. The class diagram in the appendix shows the coherence of the MAIA meta-model. Finally, the progress we have been able to make so far towards automated translation of MAIA-based models to executable simulation software indicates that the semantics of the concepts in the meta-model are well-defined.

Completeness

- 5.3 To test whether MAIA provides sufficient concepts to conceptualize a broad range of social systems, we have applied it in four case studies. The first case study was about transitions in consumer lighting, where the purpose was to identify those policies that would be effective in the transition towards more efficient lighting (i.e., LED) in households (Ligtvoet et al. 2011). The second case study involved modelling the woodfuel market in Switzerland to obtain more knowledge of the rules that govern the woodfuel market. The goal was to understand the influence of different factors (e.g., demand, fossil energy prices, co-product markets, policy measures, and natural disasters) on the availability of woodfuel, and find more successful measures to foster biofuel (Steubing et al. 2011). The third case study was the e-waste recycling sector in Bangalore, India, where we tried to identify those factors influencing the transition of the informal recycling sector in Bangalore into a system cooperating with professional end refiners which would lead to less environmental and health hazards (Ghorbani et al. 2011). Our fourth case study was about understanding those factors that influence the development of a manure-based energy system within rural regions in the Netherlands (De Korte 2012).
- 5.4 All four case studies addressed policy problems in a social system with some technical dimensions. The meta-model went through major revisions during the first two case studies, while the third and fourth case studies only led to minor changes. This suggests that the present meta-model is complete at least for the respective model-based inquiries.

Parsimony

- 5.5 Reflections on the case studies, and discussions with experts, led us to remove redundant concepts. The first version of the MAIA meta-model featured *location* as well as *physical component*. Because of the similarity in the attributes of these two concepts, the *location* concept was dropped. The concepts *group* and *interaction* proved to be redundant. *Groups* can be defined as agents, which may contain other agents. We the concept of *interaction* should not be specified because it emerges from agent actions and role dependencies.

Usefulness

- 5.6 We can assess the usefulness of the MAIA meta-model in developing an agent-based because the consumer lighting model and the wood-fuel market model were developed first without MAIA, and later redeveloped using MAIA. The feedback we received from the modellers of the four case studies in general also reflects its usefulness. Three of these modellers were other people than the MAIA developers, and they were all social scientists. They stated that it was more feasible for them to learn MAIA than programming, and that the concepts were relatively easy for them to understand. The major contribution of the meta-model was the integration of social structure (role, institutions) into the models, which had not been done before. This resulted in more diversity in agent behaviour, as the agents took different roles and (dis)obeyed different rules. The users of the first two cases indicated that the MAIA framework helped them in adding more details to the models, while the models were easier to implement and analyse. A second contribution was that the results of the conceptualization and design phase provided useful documentation, whereas for the previous models the documentation consisted of only the program code (plus some small details in publications). This made it practically impossible to regenerate the code or even get a complete understanding of the model. Although MAIA tables and diagrams mostly contain texts in natural language, it was relatively easy to translate them to computer code. This became very apparent in the latter two case studies (e-waste and bio-gas), where the analyst and the programmer were two different people communicating mainly via MAIA.



Discussion and Conclusion

- 6.1 The primary motivation for developing the MAIA meta-model was to bring ABM within the reach of a broader community of social science researchers. The second motivation for the development of MAIA was to integrate social concepts into ABM in order to facilitate the development of more realistic models.
- 6.2 Social concepts are commonly addressed in multi-agent systems (MAS) literature, but absent in most agent-based social simulation practices, as agents are typically considered to be extremely simple entities in what Epstein (2006) calls generative social science. In the ongoing debate on ABM with simple versus complex agents (Deffuant et al. 2003; Edmonds & Moss 2005), we take the position that conceptual models for agent-based social simulation should be "fit for purpose". By building on, and extending the IAD framework, the MAIA meta-model facilitates conceptualization of systems in which institutions are major determinants for social behaviour, but without imposing or excluding other mechanisms by which individuals come to act. MAIA models offer the possibility to use agents that are able to decide whether to comply with social structures (i.e., norms, rules and shared strategies; as explained by the child hiring example).

Contributions

- 6.3 MAIA contributes to the ABM practice in the following aspects:
- It visualizes conceptualization through tables and diagrams, and supports the process with the MAIA tool.
 - With some tutorial, the modeller can enter the information into the tables and diagrams without having any knowledge of computer science, programming, or even agent-based modelling.
 - MAIA provides a precise ontology for IAD concepts to facilitate ABM of various social systems.
 - The MAIA meta-model language is independent both of programming language and application domain, so that the concepts can be documented

and reused as required (as pointed out in (Heath et al. 2009)).

- MAIA facilitates collaborative model development. On the one hand, it is the means of communication between the modeller and the programmer; on the other hand, application of MAIA produces a structured representation of the modellers' perception of the system that can be shown to domain experts and stakeholders for concept verification before implementation.

6.4 Besides these intended contributions we see an additional benefit. During system conceptualization, the system boundaries become more clear because MAIA promotes the modeller to think about the concepts that may be relevant to be included in the agent-based model. In MAIA, the modeller can conceptualize a system to a great extent, and later select only a subset of the conceptualized components for the actual implementation. Therefore, rather than an implicit selection of components for implementation, this process can be done more explicitly and transparently, leading to better selection criteria. Our case studies have shown that the level of analysis with MAIA may already be sufficient to draw conclusions about the system, and thus help decide whether developing an agent-based model will indeed contribute to the understanding of the problem that needs to be addressed.

Scope

6.5 Since we embrace Ostrom's assumption that every social system has an institutional backbone, we see the MAIA framework suitable for modelling any form of social system. However, our work to date has been biased towards policy analysis. With the MAIA meta-model and the current level of implementation, agent-based models developed with MAIA are best suited for evaluating policy problems: policy instruments can be implemented as a set of institutional rules in the agent-based model.

Outlook

6.6 The concepts in the MAIA meta-model are abstract enough to be the basis for various studies including the evolution of institution, market mechanism, etc. By building custom libraries and plugins for various concepts in MAIA for different social settings, we can develop an integrated environment for ABM of social systems.

6.7 It is possible to develop libraries with specializations of various concepts in MAIA, not only for different application areas, but also libraries that expand the generic capabilities of agents. For example, pattern recognition for institution detection by agents could be implemented, enabling agents to change institutions with some predefined algorithms and rules. Another possibility would be to implement learning algorithms that allow modelling adaptive agents. We can also conceive of predefined institution libraries that provide entity actions and plans for voting, conflict resolution, negotiation, etc. We expect that all these libraries can be coded using the basic concepts defined in the MAIA meta-model.

Limitations and Future Work

6.8 This research is a first step in the process of bringing ABM within the reach of all social scientists, leaving many areas for improvement and further research. We believe that MAIA is easier to learn and use than programming and modelling for a social scientist. This can further improved by making the MAIA web-tool more user-friendly. Currently, we have facilitated semi-automatic generation of code, but our final goal is to provide software that produces runnable agent-based models from the MAIA concepts. This software may also map MAIA to Mason or Repast for running simulations that make use of the existing options and libraries these ABM platforms provide. Furthermore, conceptualization of model evaluation is an area which requires further development. The identification of outcome variables and validation variables, are only an initial step towards this requirement. Finally, the MAIA methodology has a specific focus on conceptualization and implementation of an agent-based model; data analysis, experimental design and testing are among the methodological steps which the MAIA methodology can be extended with.

6.9 In conclusion, we should point out that, like any other software development methodology, MAIA is a guideline, rather than a strict recipe. The decisions about what to model and what not to model, and how to model different aspects of a socio (-technical, -ecological) system will always require the intellectual power, judgment, creativity and originality of the modeller.



Appendix 1 The MAIA meta-model class diagram.



Notes

- 1 More information about MAIA and the web-tool can be found at: <http://maia.tudelft.nl>
- 2 Action situations and institutional statements have sufficient details to be translated into computer code compared to other concepts in the IAD.
- 3 MAIA can also be referred to as a framework. The distinction we make between a framework and a meta-model is their usage. Meta-models are more restricted in their usage because they are only intended to describe models while frameworks can be used for various purposes.
- 4 This personal value may bring less wealth to them since they would use more expensive materials for gold extraction to keep safe.
- 5 The decision making of agents is associated with the actions they perform, therefore, this concept is presented as part of the operational structure.
- 6 The MAIA software is a web-based application developed in Java Script and HTML. Several libraries including jquery and jquery-ui (for the user interface), backbone.js (for managing the data model in the browser), and RaphaelJS (for rendering graphs) were used. The server backend is used to store model data. Models are simply stored as json files.
- 7 There are two different types of inputs to the software: simple inputs (e.g., string or selections) for simple attributes such as property, type and name, and complex inputs for attributes, which are defined elsewhere in the model and associated to other concepts.
- 8 The name scope originates from the definition of scope rules by Ostrom (2005) linking actions to outcomes.



References

- ATKINSON, C. & KUHN, T. (2003). Model-driven development: a metamodeling foundation. *Software*, IEEE 20(5), 36-41. [doi:10.1109/MS.2003.1231149]
- BÉZIVIN, J. (2005). On the unification power of models. *Software and Systems Modeling* 4(2), 171-188. [doi:10.1007/s10270-005-0079-0]
- BOUSQUET, F., BAKAM, I., PROTON, H. & LE PAGE, C. (1998). Cormas: common-pool resources and multi-agent systems. *Tasks and Methods in Applied Artificial Intelligence*, 826-837. [doi:10.1007/3-540-64574-8_469]
- COSENTINO, M. (2005). From requirements to code with the passi methodology. *Agent-oriented methodologies*, 79-106, Idea Group Inc., Hershey, PA, USA. [doi:10.4018/978-1-59140-581-8.ch004]
- CRAWFORD, S. & OSTROM, E. (1995). A grammar of institutions. *American Political Science Review*, 582-600. [doi:10.2307/2082975]

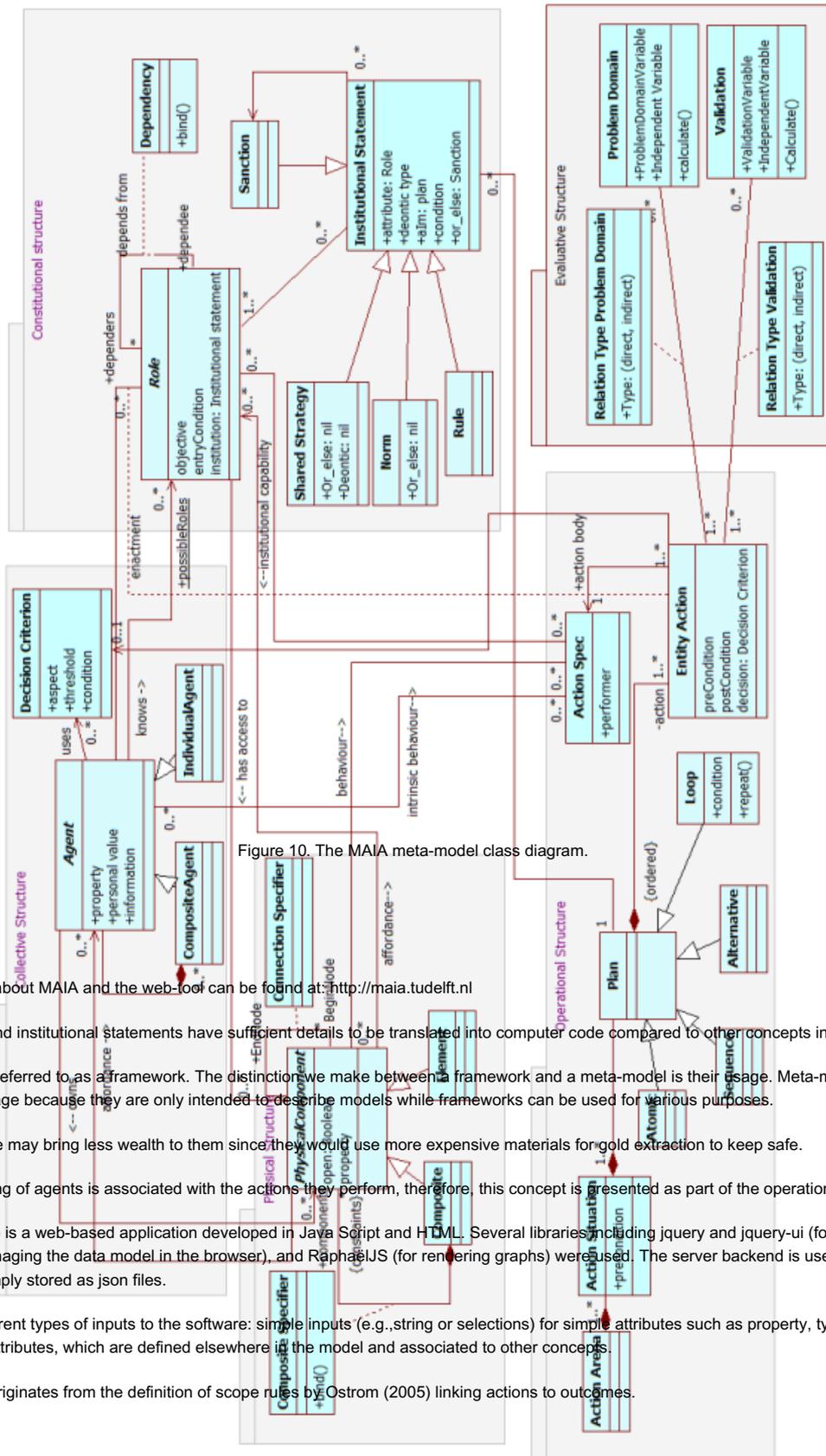


Figure 10. The MAIA meta-model class diagram.

- DEADMAN, P., SCHLAGER, E. & GIMBLETT, R. (2000). Simulating common pool resource management experiments with adaptive agents employing alternate communication routines. *Journal of Artificial Societies and Social Simulation* 3(2).
- DE KORTE, F. (2012). *Exploring the potential of manure-based energy production in Salland*. MSc. thesis. Delft University of Technology
- DIGNUM, V. (2004). *A model for organizational interaction: based on agents, founded in logic*. Ph.D. thesis. University of Utrecht
- DROGOUL, A., VANBERGUE, D. & MEURISSE, T. (2003). Multi-agent based simulation: Where are the agents? *Multi-agent-based simulation II*, 43-49.
- EDMONDS, B. & MOSS, S. (2005). From KISS to KIDS - An 'Anti-simplistic' Modelling Approach. *Multi-Agent and Multi-Agent-Based Simulation*, 130-44, Springer . [doi:10.1007/978-3-540-32243-6_11]
- DEFFUANT, G., WEISBUCH, G., AMBLARD, F. & FAURE, T. (2003). Simple is beautiful ... and necessary. *Journal of Artificial Societies and Social Simulation* 6(1),6 <http://jasss.soc.surrey.ac.uk/6/1/6.html>.
- EPSTEIN, J. (2006). *Generative social science: Studies in agent-based computational modeling*. Princeton University Press.
- FERBER, J., GUTKNECHT, O. & MICHEL, F. (2004). From agents to organizations: an organizational view of multi-agent systems *Agent-Oriented Software Engineering IV*, Springer, 443-459
- GARRO, A. & RUSSO, W. (2010). Easyabms: A domain-expert oriented methodology for agent-based modeling and simulation. *Simulation Modelling Practice and Theory* 18(10), 1453-1467. [doi:10.1016/j.simpat.2010.04.004]
- GHORBANI, A. & DIGNUM, V. & SHEORATAN, S. & DIJKEMA (2011). Applying the MAIA Methodology to Model the Informal E-waste Recycling Sector. *In: The seventh conference of the European Social Simulation Association (ESSA), Montpellier, France*.
- GIDDENS, A. (1984). *The constitution of society: Outline of the theory of structuration*. University of California Press.
- GIDDENS, A. & TURNER, J. (1988). *Social theory today*. Stanford University Press.
- GILBERT, G. & TROITZSCH, K. (2005). *Simulation for the social scientist*. Open University Press.
- GORDILLO, G. & ANDERSSON, K. (2004). From policy lessons to policy actions: Motivation to take evaluation seriously. *Public Administration and Development* 24(4), 305-320. [doi:10.1002/pad.325]
- HA, N., AGUSA, T., RAMU, K., TU, N., MURATA, S., BULBULE, K., PARTHASARATHY, P., TAKAHASHI, S., SUBRAMANIAN, A. & TANABE, S. (2009). Contamination by trace elements at e-waste recycling sites in Bangalore, India. *Chemosphere* 76(1), 9-15. [doi:10.1016/j.chemosphere.2009.02.056]
- HASSAN, S., FUENTES-FERNÁNDEZ, R., GALÁN, J., LÓPEZ-PAREDES, A. & PAVÓN, J. (2009). Reducing the modeling gap: On the use of metamodels in agent-based simulation. *In: 6th Conference of the European Social Simulation Association (ESSA) 2009*.
- HEATH, B., HILL, R. & CIARALLO, F. (2009). A survey of agent-based modeling practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation* 12(4), 9 <http://jasss.soc.surrey.ac.uk/14/4/9.html>.
- HEDSTRÖM, P. & SWEDBERG, R. (1996). Social mechanisms. *Acta Sociologica* 39(3), 281-308. [doi:10.1177/000169939603900302]
- IBA, T., MATSUZAWA, Y. & AOYAMA, N. (2004). From conceptual models to simulation models: Model driven development of agent-based simulations. *In: 9th Workshop on Economics and Heterogeneous Interacting Agents*.
- JANSSEN, M., ALESSA, L., BARTON, M., BERGIN, S. & LEE, A. (2008). Towards a community framework for agent-based modelling. *Journal of Artificial Societies and Social Simulation* 11(2), 6.
- LIGTVOET, A., GHORBANI, A. & CHAPPIN, E. (2011). A methodology for agent-based modeling using institutional analysis applied to consumer lighting. *In: Agent Technologies for Energy Systems, Tenth international conference on Autonomous Agents and Multi Agent Systems (AAMAS), Taipei, Taiwan*.
- MACAL, C. & NORTH, M. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation* 4(3), 151-162. [doi:10.1057/jos.2010.3]
- MANSON, S. (2005). Agent-based modeling and genetic programming for modeling land change in the southern yucatan peninsular region of mexico. *Agriculture, ecosystems & environment* 111(1-4), 47-62. [doi:10.1016/j.agee.2005.04.024]
- MINAR, N. (1996). *The swarm simulation system: A toolkit for building multi-agent simulations*. Santa Fe Institute Santa Fe, NM.
- NORTH, M., COLLIER, N. & VOS, J. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 16(1), 1-25. [doi:10.1145/1122012.1122013]
- OAKERSON, R. (1992). Analyzing the commons: A framework. *Making the commons work: Theory, practice and policy*, 41-59.
- OSTROM, E. (1991). *Crafting institutions for self-governing irrigation systems*. Ics Press.
- OSTROM, E. (2005). *Understanding institutional diversity*. Princeton University Press.
- OSTROM, E., GARDNER, R. & WALKER, J. (1994). *Rules, games, and common-pool resources*. University of Michigan Press.
- PICARD, G. & GLEIZES, M. (2004). The adelfe methodology. *Methodologies and Software Engineering for Agent Systems*, 157-175. [doi:10.1007/1-4020-8058-1_11]
- SANSORES, C. & PAVÓN, J. (2005). Agent-based simulation replication: A model driven architecture approach. *MICAI 2005: Advances in Artificial Intelligence*, 244-253. [doi:10.1007/11579427_25]
- SCHARPF, F. (1997). *Games real actors play: Actor-centered institutionalism in policy research*. Westview Press.
- SCHMIDT, D. (2006). Guest editor's introduction: Model-driven engineering. *Computer* 39(2), 25-31. [doi:10.1109/MC.2006.58]
- SHEORATAN, S. (2011). *Agent Based Modeling the e-waste recycling sector in Bangalore*. M.Sc. Thesis, Delft University of Technology, Leiden University.
- STUEBING, B., KOSTADINOV, F., GHORBANI, A., WAGER, P., ZAHA, R., THEES, C. & LUDWIG, C. (2011). Agent-based modeling of a woodfuel market' factors that affect the availability of woodfuel. *In: ISIE'11, Berkeley, U.S.A.*

- TISUE, S. (2004). Netlogo: Design and implementation of a multi-agent modeling environment. *In: In Proceedings of Agent 2004*. Citeseer.
- WINOGRAD, T., BENNETT, J., DE YOUNG, L. & HARTFIELD, B. (1996). *Bringing design to software*. ACM Press, New York, NY.
- WYNNE, S. (1989). *The Land boards of Botswana: A problem in Institutional Design*. Ph.D. Thesis, Indiana University.
- YANDLE, T. & DEWEES, C. (2003). Privatizing the commons... twelve years later: fishers' experiences with New Zealand's market-based fisheries management. *The Commons in the New Millennium: Challenges and Adaptations* , 101-27.