

## Amineh Ghorbani Structuring Sociotechnical Complexity

Modelling Agent Systems Using Institutional Analysis

61



NEXT GENERATION INFRASTRUCTURES FOUNDATION

## Structuring Socio-technical Complexity

Modelling Agent Systems using Institutional Analysis

## Structuring Socio-technical Complexity

Modelling Agent Systems using Institutional Analysis

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Delft, op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben, voorzitter van het College voor Promoties, in het openbaar te verdedigen op vrijdag 14 juni 2013 om 10:00 uur door

Amineh GHORBANI

Master of Science in Information Technology Engineering, University of Tehran geboren te Isfahan, Iran. Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M.P.C. Weijnen

Copromotor: Dr. ir. G.P.J. Dijkema

Copromotor: Dr. ir. M.V. Dignum

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. M.P.C. Weijnen	Technische Universiteit Delft, promotor
Dr. ir. G.P.J. Dijkema	Technische Universiteit Delft, copromotor
Dr. ir. M.V. Dignum	Technische Universiteit Delft, copromotor
Prof. dr. C. Sierra	Spanish Research Council & Sydney Technical University,
Prof. dr. S. Moss	University of Koblenz-Landau,
Prof. dr. O. Boissier	Ecole Nationale Suprieure des Mines of Saint-Etienne,
Prof. dr. ir. T.A.J. Toonen	Technische Universiteit Delft
Prof. dr. ir. P.M. Herder	Technische Universiteit Delft (reservelid)

ISBN 978-90-79787-50-0

*Published and distributed by:* Next Generation Infrastructures Foundation P.O. Box 5015, 2600 GA Delft, The Netherlands

Phone:	$+31\ 15\ 278\ 2564$
Fax:	$+31\ 15\ 278\ 2563$
E-mail:	info@nextgenerationinfrastructures.eu
Website:	http://www.nextgenerationinfrastructures.eu

This research was funded by Delft University of Technology and the Next Generation Infrastructures Foundation.

Keywords: social simulation, agent-based modelling, institutional analysis, policy, socio-technical system, model-driven software development, conceptualization

Copyright © 2013 by A. Ghorbani

Some rights reserved. This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Netherlands License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/nl/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Cover: Kim Berrywood (licensed copy) & Purevil (http://crazyslug.net/) Printed in the Netherlands by Gildeprint on G-print FSC Mixed Sources Available on http://repository.tudelft.nl

E-mail: amineh.ghorbani@gmail.com Website: http://aminehghorbani.weblog.tudelft.nl/

## Contents

Li	st of	Figures	ix
Li	st of	Tables	xi
Li	st of	Abbreviations	xiii
A	ckno	wledgements	xv
Ι	ES	TABLISHING THE RESEARCH FRAMEWORK	1
1	Inti	roduction	3
	1.1	Motivation	3
	1.2	Research Theme	5
	1.3	Research Approach	6
	1.4	Scope	9
	1.5	Outline	10
<b>2</b>	The	eoretical Foundations	11
	2.1	Introduction	11
	2.2	Social Science Foundation	12
	2.3	Computational Foundation	18
	2.4	Conclusion	23
тт	C	ONCEPTUALIZATION OF AGENT-BASED MOD-	
E	LS		<b>25</b>
3	MA	IA	27
	3.1	Introduction	27
	3.2	Describing a Typical Socio-technical System	28
	3.3	The MAIA Meta-model	30
	3.4	Using MAIA	38
	3.5	Evaluating the MAIA Meta-model	47
	3.6	Compliance with the IAD Framework	51
	3.7	Soundness of MAIA	54
	3.8	Conclusion	55

Cas	e Study Evaluation	57
4.1	Introduction	57
4.2	Case Study I: Consumer Lighting Transitions	58
4.3	Case Study II: The Wood-Fuel Market	66
4.4	Case Study III: E-waste Recycling	72
4.5	Case Study IV: Manure-based Bio-gas Production	84
4.6	User Survey Evaluation	94
4.7	Conclusion	96
	Case 4.1 4.2 4.3 4.4 4.5 4.6 4.7	Case Study Evaluation4.1Introduction

#### III FORMALIZED INSTRUMENT FOR AGENT-BASED MODELLING 97

<b>5</b>	Mo	del-driven Development for MAIA	99
	5.1	Introduction	99
	5.2	Agent-based Simulation Architecture	100
	5.3	A Model-driven Approach to Build Simulations	110
	5.4	From Model to Simulation	112
	5.5	Platform Independence for MAIA	115
	5.6	Conclusion	121
6	For	mal Specification of MAIA	123
	6.1	Introduction	123
	6.2	MAIA Formal Definition	124
	6.3	ADICO Semantics	131
	6.4	Conclusion	143

#### IV INSIGHTS

7	An	Evaluation Framework for ABMS Platforms	147
	7.1	Introduction	147
	7.2	Tools and Methods for ABMS	148
	7.3	A Framework to Compare and Evaluate ABMS Platforms	151
	7.4	Feature-based Evaluation of ABMS Tools	156
	7.5	Comparison between the Four Tools	159
	7.6	Conclusion	161
8	Dis	cussion and Conclusion	163
	8.1	Overview	163
	8.2	Development Process of MAIA	164
	8.3	Research Outcomes	165
	8.4	Reflection	169
	8.5	Lessons Learnt	171
	8.6	Future Work	172

#### Appendices

145

A Evaluation Form for Usability and Usefulness of MAIA	177
B MAIA Syntax	179
C Evaluation Form for ABMS Platforms	181
Bibliography	185
Summary	199
Samenvatting 2	
Curriculum vitae	
NGInfra PhD thesis series on infrastructures 2	

## List of Figures

1.1	The research process	8
2.1 2.2 2.3	The IAD framework (Ostrom et al., 1994)	14 18
2.4	The OperA Overall Architecture	21 23
3.1	The Collective Structure in the MAIA meta-model.	32
3.2	The Constitutional Structure in the MAIA meta-model.	34
3.3	The Physical Structure in the MAIA meta-model.	35
3.4	The Operational Structure in the MAIA meta-model.	37
3.5	The Evaluative Structure in the MAIA meta-model.	38
3.6	The UML class diagram for the MAIA meta-model	39
3.7	A worker agent card is being created in the MAIA web-based application.	40
3.8	Two institution cards for the e-waste system in the MAIA tool.	41
3.9	Objective dependencies between roles in the e-waste example	41
3.10	The hire entity action card for the e-waste example in the MAIA tool.	42
3.11	The action arena for the e-waste example in the MAIA tool	42
3.12	Part of the validation matrix for the e-waste example in the MAIA tool.	43
3.13	Participatory ABMS and separation of tasks using MAIA	47
3.14	The internal structure of action situations in the IAD framework (Ostrom 2005)	53
2 1 5	The internal structure of action situations in MAIA	53
0.10	The internal structure of action situations in MAIA	99
4.1	Operational environment for a consumer lighting model	64
4.2	The market situation illustrating the negotiation process. The dotted	70
4.3	The trading relation between different agent types. The circles, triangles and squares represent different agents types. Arrows indicate the direc- tion of the trade from the seller to the buyer (Kostadinov et al., 2012).	70
	• • • • • • • • • • • • • • • • • • • •	71
4.4	Composition diagram	77
4.5	Standard deviation over the different runs in the e-waste recycling case	
	study (Sheoratan, 2011)	82
4.6	Decision to produce green gas (De Korte, 2012)	86

4.7	Green gas production without cooperation (De Korte, 2012)	93	
4.8	Green gas production with cooperation (De Korte, 2012)		
5.1	An agent-based simulation architecture.	101	
5.2	The space module in a MAIA simulation forms around the action arena.	107	
5.3	Simulated environment using MAIA. The dotted circles represent the plans within each action situation. An arrow between two plans shows a		
	sequence. The arrow with circles on both ends represents an alternative.	108	
5.4	The main model in a MAIA simulation.	109	
5.5	Simulation results from the Family Care example comparing different		
	policies.	110	
5.6	Model-driven Software Development	111	
5.7	Platform Independent Modelling in MDSD	112	
5.8	Example of code generation using JET templates and MAIA models	113	
5.9	MDSD using AMF	116	
5.10	The AMF Meta-model - Key Collaborations (AMP, 2013)	117	
5.11	The four types of AMF Acts and their attributes (AMP, 2013)	119	
5.12	An Entity Action as a flow in AMF. The second line in each box shows		
	which components of the AMF concept we are using.	120	
7.1	Requirements for the modelling language in an agent-based simulation		
	platform	152	
7.2	Requirements for the simulation development process in an agent-based		
	simulation platform	153	
7.3	Pragmatic requirements for an agent-based simulation platform. $\ldots$ .	153	
7.4	Agent-oriented requirements in an agent-based simulation platform	154	
7.5	Simulation requirements in an agent-based simulation platform	155	

## List of Tables

3.1	Different types of institutional statements in the e-waste example	34
3.2	Concepts that map between the IAD and MAIA framework.	51
11	The Institution table for consumer lighting	62
4.1	The weights used in the decision making of agents based on their personal	02
4.2	values (Kostadinov et al. 2012)	68
43	The Institution table for the e-waste recycling system	76
4.4	Part of the role enactment table for the e-waste recycling system	79
4.5	Scope matrix for the e-waste recycling model.	80
4.6	Valiation matrix for the e-waste recycling model.	80
4.7	Analysis of 5 areas in the t-SNE graphs (Sheoratan, 2011).	82
4.8	MCDM factor design	87
4.9	The Institution table for the bio-gas energy system	88
4.10	Scope matrix	90
4.11	Validation matrix	91
4.12	Usefulness of MAIA for developing the four case studies and its usability.	95
5.1	Mapping between MAIA concepts and simulation modules	102
6.1	Examples of ADICO institutional statements in the family care setting.	128
6.2	Examples of Behaviours that can be assumed shared strategies	134
6.3	Strategy Types	134
6.4	Examples of Shared Strategies	139
6.5	Concepts related to shared strategy in current MAS literature	142

## List of Abbreviations and Acronyms

ABM	Agent-based model
ABMS	Agent-based modelling and simulation
ACI	Actor-centred institutionalism
ADICO	Attribute, deontic, aim, condition, or else
AHP	Analytical hierarchy process
$\mathbf{AMF}$	Agent modelling framework
AOSE	Agent-oriented software engineering
CFL	Compact fluorescent lamp
CIM	Computational independent model
CRI	Colour-rendering index
IAD	Institutional analysis and development framework
JET	Java emitter template
LED	Light-emitting diode
MAIA	Modelling agent systems using institutional analysis
MAS	Multi-agent systems
MCDM	Multi-criteria decision making
MDSD	Model-driven software development
MES	Manure-based energy system
OCL	Object constraint language
OM	Organizational model
$\operatorname{PER}$	Professional end refiner
PIM	Platform independent model
$\mathbf{PSM}$	Platform specific model
$\mathbf{SM}$	Social model
SS	Social simulation
SE	Software engineering

### Acknowledgements

This dissertation is the result of three and half years of research with a group of remarkable people. I thank God for giving me this unique and unforgettable opportunity and allowing me to enjoy every minute of it. And, I thank every single person, colleague, friend or family that I have come across during this journey. These relationships, even if really small, have positioned me at my current status in life.

I am indebted to prof. Margot Weijnen, my promoter, for creating this social and friendly research environment. I would like to thank her for her broad scientific vision and insightful comments which helped me complete this manuscript.

I am truly honoured and grateful for having dr. Gerard Dijkema as my copromotor. He showed me what true supervision means, by giving me the flexibility to carry on with my ideas while keeping me within the context and boundaries of this research. His kindness and support gave me confidence and motivated me to stay focused on the meandering path of a PhD.

I have also been extremely lucky to have dr. Virginia Dignum as my second copromotor. She showed me that there is not end to how helpful a person can be. Her smart ideas made this research a true joy and her amazing network of researchers allowed me to acquire support whenever and wherever I required.

I owe special thanks to the members of my defence committee. I appreciate all for accepting to be in my committee and providing feedback. I am especially grateful to prof. Scott Moss for giving such thorough feedback on this dissertation.

This research would have been impossible without the help and support of many other great scientists whom I have been honoured to meet and work with during these years. I thank dr. Igor Nikolic for making ABM such a prominent topic of research in this faculty and allowing me to continue his line of research. Without his encouragement and belief in my potential, I would have not been able to think out of the box. I am especially privileged to have worked with dr. Pieter Bots. Besides his remarkable contributions to the development of MAIA, he has been the best teacher I have ever had. I am also honoured to have visited dr. Pablo Noriega. The time he spent with me discussing what I thought I knew, allowed me to truly understand the meaning of my research and the actual power of 'ADICO'. I thank dr. Huib Alderwereld for his 'formal' contribution and for helping me operationalize my theoretical research. I am also delighted for the insights dr. Gert Jan Hofstede, prof. Catholijn Jonker and dr. Frank Dignum have provided for me and I am very much looking forward to continuing my research with them.

I am indebted to all those who helped me evaluate my work by putting it to practice. First of all, I thank Fabian Kostadinov and Bernhard Steubing for putting the initial ideas of this research to test. I really enjoyed the intensive modelling work I pursued with them. I thank the students who spent time and effort to apply MAIA to their research: Sathyam Sheoratan, Femke de Korte, Noortje Schrauwen and Reinier Verhoog. I enjoyed working with everyone of them and also Ahmad Reza Mir Mohammadi who is performing a different type of interesting and challenging ABM research. The final student that I would like to give special thanks to is Bert Belder for building the MAIA web tool and trying to improve it everyday.

Research at TPM would have been colourless without the great colleagues and friends I found during this time. Rob Stikkelman has been the most humorous yet insightful person who filled me up with information about everything. Andreas Ligtvoet was a true joy to work and talk with. Thank you for all your feedback especially on my propositions. Emile Chappin, Martti van Blijswijk and Andrew Bollinger, travelling with you has left fun and unforgettable memories. Chris Davis, you are *the* person who is there whenever anyone needs help. Without you, I would not have been able to make sense out of any data. Reinier van der Veen thank you for being such a calm and pleasant friend. Behzad Behdani, I appreciate your deep scientific and moral advices. Inge Claeys, thank you for showing me various dimensions of the Dutch culture. Finally, my wonderful room mates, Remco Verzijlbergh and Chang Yu, you have enlightened my daily atmosphere so much that I looked forward to coming to the office every day. Furthermore, I would like to thank Chang and Inge for accepting to be my paranymphs in order to form a girls only defence team!

Last but not least, my biggest appreciation goes to my brilliant husband, Yashar. He could not have supported me more during my PhD and also during our life together. I feel so fortunate to have you by me and I can not imagine life without you. I am also truly grateful for the close circle of friends I have found in the Netherlands which have helped me overcome the feeling of being far from home. Finally, I would like to thank my family especially my parents who provided the opportunity for me to grow and excel in what I am best at and have always supported my decisions throughout my life.

Amineh Ghorbani - May 2013

## Part I

## ESTABLISHING THE RESEARCH FRAMEWORK

Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.

Albert Einstein

# Introduction

#### 1.1 Motivation

How would taxation on light bulbs or subsidies on LED lamps influence the behaviour of consumers towards more energy saving habits? Can investment on manurebased biogas systems improve farming prospects for animal farmers? And, does fining recyclers in a developing country prevent them from hiring children and using dangerous chemicals when they are recycling electronic appliances?

The similarity of these questions is not their domain but the specific type of problem they are dealing with. These questions all arise from socio-technical systems, i.e., social systems that are intertwined with technology. Social structures (e.g., norms, rules and culture) play a major role in shaping these systems. Furthermore, the questions all address policy problems, exploring the long term effect of strategic decisions on the operational behaviour of individuals and on the global outcomes of the system.

Socio-technical systems are complex. They consist of heterogeneous decision making entities and technological artefacts. These systems are governed by public policy in a multi-scale institutional context. For defining effective policies, an understanding of the system is gained through various approaches, ranging from benchmarking and historical analysis (Scharpf, 1997) to computational simulations (Gilbert, 2004).

Social scientists use simulations to analyse socio-technical problems and explore policy alternatives. Simulations address complexity. They facilitate the understanding of relationships between events and explain how certain behaviours and interactions result in emergent outcomes in socio-technical systems. Furthermore, they allow the identification of desired and undesired social and technological behaviours in a system. This is especially valuable for those decisions and policies that cannot be tested on the actual system due to cost or safety issues. Another advantage of using simulations is the possibility to model and explore situations that do not or cannot exist in the real world (Gilbert, 2004).

One simulation approach that is particularly insightful to study socio-technical systems is agent-based modelling and simulation (ABMS). ABMS provides a natural representation of a social system by considering heterogeneous entities, called agents, as the building blocks of the simulation. In these simulations, agents commonly represent people, companies, governments, technological artefacts and other 'self-contained' entities. These entities interact with each other and with the environment, causing global behaviours, patterns and structures to emerge from the simulated system.

An issue with current ABMS practices follows the arguments that suggest individualism, as in ABMS, cannot explain many complex phenomena if social structures are absent in the simulation (Conte et al., 2001). Social structures emerge from individual behaviour and social interaction. However, to put agents in the context of a socio-technical system, a primary definition of the system including social structures such as cultures, norms and networks is required. This initial context would allow the decision making entities to react accordingly and in turn affect and evolve the structure.

In current ABMS, social structures are either not considered or are modelled as part of the agents. Modelling social structures within agents is far from reality because these structures are observed as independent concepts, separate from individuals in social systems. The primary consequence of simulating the combination of the two as one entity is that we would not be able to model global changes in these structures and observe how they evolve and perish, and how new structures emerge as a result of social process. Furthermore, when social structures are modelled within agents, studying their influence on individual behaviour and the system as a whole is not straightforward. More specifically, if the purpose of the simulation is to explore policy alternatives, being able to model policies as purposive design of social structure is highly instrumental for studying their effects on individuals and the system as a whole.

There are also practical drawbacks for current ABMS practices. First, compared to other simulations, agent-based models are relatively complex to build, requiring substantial programming knowledge (Railsback et al., 2006). However, the actual users of the ABMS approach are social scientists and policy makers who may have little familiarity with computational tools (Pavon et al., 2008). Second, besides the difficulties in building simulations, it is also impractical to involve various parties such as problem owners and domain experts in the simulation process. This, however, is a necessary requirement for gaining a better understanding of the system and the problem at hand (Ramanath and Gilbert, 2004).

A number of scientists advocate participatory ABMS and provide guidelines and methods for that purpose (e.g., (Ramanath and Gilbert, 2004; Chu et al., 2012; Nikolic, 2009; Becu et al., 2008)). Although these methods are well developed, defining methods of stakeholders involvement from early conceptualization to the final communication of results, with current ABMS tools, it appears to be rather difficult to actually involve different parties in the simulation *process* (Becu et al., 2008). Currently, the outcome of model specification can only be shown to the problem owners once the simulation or its prototype have been implemented. However, an early verification of the concepts before implementation not only reduces the costs of making changes in the simulation, but it also increases model reliability through experts validation. Due to the difficulties in participatory model development, ABMS is currently *not* being recognized or appreciated as one of the most powerful analysis tools despite its high potential (Ramanath and Gilbert, 2004).

ABMS is an insightful tool for studying socio-technical systems. However, to really understand and link various levels of behaviour in these systems and increase the usability of ABMS, we need to overcome the aforementioned conceptual and practical limitations of this approach.

#### 1.2 Research Theme

To govern socio-technical systems and develop effective policies, analysts require an understanding of the systems and a test bed to explore and compare policy alternatives. ABMS can be used as an insightful tool to address this issue. However, there are various limitations regarding ABMS that need to be overcome in order to augment the insights and increase the usability and applicability of this exploratory approach. These considerations lead to the following research question and subquestions:

#### 1.2.1 Research Question

How can we build social structures in agent-based models and increase the utility of ABMS for policy analysis?

- 1. Which concepts and relations define social structures in agent-based models?
- 2. How are social structures connected to other aspects of a socio-technical system?
- 3. How can an ABMS tool be made more accessible to users with less or no experience in programming or simulation?
- 4. How can we facilitate participatory model development?

#### 1.2.2 Objectives

In order to provide a set of concepts and relations that define social structures for ABMS, we can develop a modelling framework that decomposes these structures into a set of components. However, social structures are not isolated; they influence entities in socio-technical systems and are influenced by them. Therefore, we must extend the modelling framework to include those components of the system that

may be linked to social structures in a socio-technical system. This would include social, physical and environmental entities.

Ultimately, to *integrate* social structures into agent-based models, a comprehensive conceptual framework is required that decomposes a socio-technical system into a set of *concepts* and *relations*. Therefore, we combine the first two sub-questions into one objective as follows:

1. To develop a conceptual framework for describing a socio-technical system and formalize it for building computer simulations.

To answer the practical sub-questions, we set the next two objectives:

- 2. To build a tool and provide simulation development guidelines for social scientists and policy makers with different levels of expertise in programming and simulation.
- 3. To enable participatory ABMS from the early conceptualization phase of the simulation process.

#### 1.3 Research Approach

In order to gain insights into socio-technical systems, we will build an ABMS framework that is on the one hand conceptually rich in representing social structures and other components of a socio-technical system, and is on the other hand methodologically useful and applicable for its users.

We will first build a conceptual framework to decompose and structure a sociotechnical system with an agent-oriented perspective. Besides conceptualizing social structures, the framework should capture and explain individual (i.e., agent) behaviour, characteristics and decision making, and define their relationship with social structures. In addition, other components of the system may also be related to these structures. For example, the implementation of a policy as a form of social structure may be through the installation of a technical artefact such as a gate or a CCTV camera.

The conceptual framework should be defined in a high level language to be understandable to non-programmers. Therefore, to build simulations using this conceptual framework, we will use a method to transform a model described in a high level language into an executable simulation, coded in low level programming languages.

As Ramanath and Gilbert (2004) discuss, 'A predefined modelling structure, organisation of the simulation and a certain degree of formality of activities would motivate actors to participate in the development, trust the simulation and use its outputs'. Therefore, the modelling framework will be an initial step in enabling participatory model development. During the development of this framework, we will also identify the potential users and their forms of interaction. These users include domain experts, policy makers, problem owners, modellers and programmers.

#### 1.3.1 ABMS as the Simulation Approach

Computer simulation is a well established field of research at the intersection between social, mathematical and computer sciences (Conte et al., 1998). ABMS is especially promising for this research because it facilitates the construction and exploration of 'artificial societies of autonomous agents' (Conte et al., 1998; Bankes, 2002).

In ABMS, agents are decision making entities which are able to make autonomous decisions and act and react on their environment and on other agents (Gilbert, 2004). The society is built from bottom-up, resulting in emergent behaviours, patterns and structures as a result of individual interaction.

The most fundamental reason for selecting this simulation approach among other approaches such as differential equations and statistical modelling is that these competing tools impose restrictive or unrealistic assumptions such as linearity, homogeneity, normality and stationarity (Bankes, 2002). While these assumptions may provide insights for some problems, they may affect the reliability of the models for others. Besides having an individual-based perspective that provides a more natural representation of socio-technical systems, ABMS has less of these restrictive assumptions (Bankes, 2002).

One other important reason for choosing ABMS for social sciences is the power to demonstrate emergent phenomena at system level. This is especially required for policy problems where the goal is to influence individual behaviour in order to evolve system properties (Conte et al., 2001).

#### 1.3.2 Institutional Analysis for Describing Socio-technical Systems

In social systems, institutions are sets of rules that structure social behaviour and interaction (Hodgson and Calatrava, 2006; North, 1990; Ostrom et al., 1994). Institutions are more elicitable and tangible than human behaviour and reduce empirical variance (Scharpf, 1997). Therefore, besides providing social structure, incorporating institutions into agent-based models helps develop more tangible assumptions about agent decision making and behaviour because we can take the influence of institutions in enabling or restricting those behaviour into account. However, to maintain the explanatory power of institutions, we must also take the flexibility of agent decision making regarding institutions into account. In other words, agents may decide not to comply with a certain institution, considering the payoff for noncompliance.

Institutional analysis is commonly used to study socio-technical systems especially in the field of institutional economics (cf. (North, 1990; Scharpf, 1997; Ostrom, 2005; Williamson, 1998)). The institutional analysis and development framework (IAD) by the Nobel Laureate Ostrom, describes various components of a sociotechnical system and explains how they are related to institutions (i.e., social structure). Therefore, IAD is in line with our research objective and an appropriate starting point to design the modelling framework.

Although IAD addresses institutions and other components of a socio-technical system, it does not have enough computational rigour and details to formulate the simulation of a socio-technical system. Therefore, we will be in search of other theories and frameworks for those aspects of the system, where the level of detail in IAD does not satisfy our purpose.

#### 1.3.3 Model-driven Development

Model-driven software development (MDSD) is a field of research in software engineering that facilitates the development of software systems from high level languages. The framework we will be developing is described in a high level language that can not directly result in executable computer simulations. Therefore, we will be using this approach to transform high levels of abstraction to computer interpretable languages.

There are two requirements for MDSD: a modelling language to make an agentbased conceptual model, and specification of transformation procedures to produce executable simulations from models described in a high level language. Therefore, the modelling framework we will be developing, will be formulated as a high level modelling language with additional transformation protocols.

The overall research process is illustrated in Figure 1.1. We will select a number of social theories and frameworks that together provide an overall description of a socio-technical system with an institutional perspective. We will combine the selection of these social theories to develop a framework which will at the same time be presented as a meta-model for building agent-based computer simulations. Besides the theoretical research to develop the framework, we will also use case studies during the development process of the framework. These case studies will eventually also be used to evaluate the framework and its accompanying tools and method. The evaluation process will continue by comparing the outcome of this research with related work. This will also lead to the concluding remarks of the thesis.



Figure 1.1 – The research process.

#### 1.4 Scope

#### 1.4.1 Scientific Relevance

This is a multidisciplinary research that aims to bridge the gap between the theoretical analysis of socio-technical systems and agent-based social simulation. By identifying the abstract concepts and relations in a socio-technical system, we give theoretically defined structure and content to agent-based models which are currently developed as a collection of simple agents. A formalization of these concepts and relations bridges the gap between social and computational sciences as concepts defined in a high level social science language would have a computational representation and would thus be usable in simulation. This is of mutual benefit for both sciences since the computational side can capture more of a socio-technical system in the artificial system, and the social scientists can use computational power to explore more possibilities and outcomes in their decision making process.

#### 1.4.2 Contribution

The contributions of this research can be divided into three areas:

Agent-based Modelling and Simulation This research will add to the ABMS research by providing a conceptualization framework (cf. (Railsback et al., 2006)) that decomposes and structures various aspects of a socio-technical system into a set of concepts and relations that can be used to make an agent-based simulation. It further contributes to this line of research by facilitating simulation development from high level model descriptions. Finally, this research provides an environment for participatory model development.

**Policy Analysis** This research contributes to the policy analysis domain by presenting a tool to gain insights into socio-technical systems. Participatory model development facilitates participatory decision making which is another contribution to policy analysis.

Multi-agent systems research Multi-agent systems (MAS) are software systems that serve a specific purpose in an environment. For example an electronic auction or a traffic system may be implemented as multi-agent systems. This line of research is different from ABMS in the sense that simulations are used for exploratory purposes and are not used to perform operational tasks like the mentioned examples. Nonetheless, for both disciplines, artificial societies are developed and therefore social structures are required. This research contributes to MAS research by introducing institutions and institutional analysis to this domain. The ADICO structure as part of the IAD framework will introduce new concepts and structures which can be highly instrumental for building artificial societies in MAS research.

#### 1.4.3 Audience

This research addresses various audiences in social and computational disciplines. To increase readability, we divided the manuscript into four parts. Part I of this thesis which aims at all audiences, presents the theoretical and computational foundatiosn of this research.

Part II, which presents the core contribution of this research, is aimed at social scientists and policy makers who have the intention to build agent-based models but do not have computational knowledge. This part is about 'what' to model rather than 'how' to model and the goal is to show social scientists and policy makers how an agent-based model can be conceptualized using high level concepts.

The third part of the thesis is more technical, addressing how a model conceptualized in a high level language can be translated into a computational simulation. Therefore, this part requires technical knowledge in computer science and is geared towards social and computational scientists who are more familiar with programming, software engineering and artificial intelligence.

Finally, Part VI is aimed at all audiences, explaining the evaluation process of the research and concluding this manuscript.

#### 1.5 Outline

The structure of this manuscript is as follows. In Chapter 2, we present the theoretical background of this research. We introduce and justify our selection of social theories and frameworks that we will be using to build an ABMS tool. We also explain the computational background we will be using to develop the simulation tool.

In Chapter 3, we define our ABMS framework (i.e., modelling language) that is built from the theories in Chapter 2. We will also explain how we are going to evaluate this framework and will partially evaluate it. The evaluation procedure continues in Chapter 4, where we discuss the case studies we developed using the ABMS tool. In this chapter, we also reflect on the users feedback.

In Chapter 5, we explain how the modelling framework presented in Chapter 3 can be used to build simulations, possibly automatic. In Chapter 6, we continue this discussion by presenting formal definitions, syntax and semantics that accompany our modelling framework. In Chapter 6, we also reflect more on our contribution to MAS research.

In Chapter 7, we present an overall evaluation of our modelling platform by comparing it to related research. We present an evaluation framework and use it to compare several ABMS platforms. In the final chapter, we conclude by reflecting on our findings, discussing the contributions and giving directions for future research. By three methods we may learn wisdom: First, by reflection, which is noblest; Second, by imitation, which is easiest; and third by experience, which is the bitterest.

Confucius

## 2

## **Theoretical Foundations**

#### 2.1 Introduction

In the social sciences, there are many analytical theories and frameworks that explain social phenomena (e.g., cooperation) or describe concepts in social systems (e.g., institution, agency). To get an overall description of a socio-technical system however, it is difficult to find one framework or theory that is comprehensive and at the same time provides sufficient formalities and details to structure the simulation of a socio-technical system. Therefore, to address this issue, we need to select a set of theories and frameworks that in combination provide a comprehensive and sufficiently detailed definition for a socio-technical system. In this respect, we have to be aware that different theories may have conflicting perspectives and assumptions, and therefore, may not be suitable to be combined with each other.

After choosing our selection of social theories, the next issue is to use the most appropriate simulation approach that is in line with our description of a sociotechnical system. We would also need a transformation method, to actually build the simulation from the theoretical description.

The goal of this chapter is to explain the theoretical backbone of this multidisciplinary research in detail. We explain the social theories and frameworks that we will be using to develop a conceptual framework in Section 2.2. We will then introduce the simulation approach we will be using and the method we will be taking to build the simulation in Section 2.3.

#### 2.2 Social Science Foundation

To perform an analysis of a social system, there are many theories and frameworks<sup>1</sup> in the social sciences. As discussed in Chapter 1, we will use institutional analysis as a perspective to study social phenomena. Among the institutional frameworks, the Institutional Analysis and Development framework (IAD) (Ostrom et al., 1994) is one of the most comprehensive and recognized tools. It provides a clear definition for institutions and specifies the connection between this social concept and other aspects of a socio-technical system. While taking an institutional perspective to analyse social system, IAD views the system from bottom-up, taking participants and their attributes into account. However, IAD does not provide sufficient details for making a simulation in some of its aspects. Therefore, we will use other social frameworks and theories to reach a more detailed description of the concepts that the IAD offers. The combination of these theories and frameworks willprovide a descriptive language that would help us develop a theoretically disciplined reconstruction and simplification of socio-technical systems. Our selection is based on the following criteria:

**Conceptually complementary**: All the theories and frameworks are conceptually compliant with the IAD while adding more detail to concepts that have less specifications in the IAD.

**Individual-based.** All the theories and frameworks view social systems from bottom-up reflecting the idea that it is the individuals who shape and influence a social system and are in turn affected by it.

**Institutional perspective** All the theories and framework take into account the fact that institutions, as the set of rules, shape individual behaviour and vice versa.

In this section we introduce our selection of social theories and framework.

#### 2.2.1 Frameworks for Institutional Studies

As explained in Chapter 1, we decided to take an institutional perspective to describe socio-technical systems because institutions can give social structure to agent-based models (Ghorbani et al., 2010). Furthermore, while understanding and explaining individual behaviour is complex, social rules or institutions are more elicitable (Scharpf 1997) and hence more readily identified and captured for building simulations. In this section we explain our definition of institutions in more detail and introduce two of the more applied frameworks in institutional studies, namely: IAD and Actor-centred Institutionalism (ACI).

 $<sup>^{1}</sup>$ A framework is a nested set of variables with potential relationships between them used as a common language (Ostrom, 2005), allowing the analysts to use various theories where relevant and useful for the problem (Ostrom et al., 1994). A theory provides assumptions about specific components of a framework to enable the diagnosis of a phenomenon, describe the related events and reflect on outcomes (Ostrom, 2005).

#### Institutions

The term institution has become prevalent in the social sciences in recent years. This reflects the growth in the use of the institution concept in several disciplines, including philosophy, management, sociology, politics and specifically institutional economics (Hodgson and Calatrava, 2006).

In institutional economics, institutions are defined as the set of devised rules to organize repetitive activities and shape human interaction (Ostrom, 1991; North, 1990). These rules include laws, regulations, social norms, and shared strategies amongst others. Rules are created either through an evolutionary process or purposive design. They can be called institutions only if they are accepted by those involved, used in practice, and have a certain degree of durability (Koppenjan and Groenewegen, 2005).

The durability of an institution and its frequency of change partly determine the layer of social analysis (Williamson, 1998; Ostrom, 2005). Williamson (1998) defines four layers of social analysis. At the lowest layer, operational rules are continuously changing. The agreements and contracts (i.e., governance structures) change between 1 to 10 years. The formal laws and regulations (i.e., institutional environment) change between 10 to 100 years. Finally, at the highest layer of analysis where the informal institutions, norms and culture are embedded, changes occur every 100 to 1000 years.

Social analysis can be conducted with the intention to change institutions. If institutions cause biased power relations and fail to fulfil stability or to enable decision making, there are grounds for institutional change or in other words, institutional (re)design (Klijn and Koppenjan, 2006).

Institutional (re)design refers to the devising of new social arrangements, by examining existing arrangements and altering them when necessary (Pettit, 1996). I.e., institutional redesign refers to deliberate changes in institutional characteristics. Once institutions are established, reforming them will be costly even if the circumstances that originally justified them no longer exist (Scharpf, 1997). Therefore, in order to (re)design institutions, one should gain a thorough understanding of the institutional rules, their settings and their origin.

#### Institutional Analysis and Development Framework

The Institutional Analysis and Development (IAD) framework was originally designed by Oakerson (1992) to analyse the problem of the commons but developed as a more general institutional framework throughout the years. IAD addresses the different components of a socio (-technical, -ecological) system that need to be analysed for institutional (re)design (Ostrom, 2005).

The IAD decomposition of a social system is presented in Figure 2.1. This framework is an institutional-driven tool for (1) understanding the underlying structures of a social system (illustrated on the left side of Figure 1), (2) capturing the operational environment (at the centre of Figure 1), and (3) observing the patterns of interaction and outcomes, given a set of evaluation criteria (depicted on the right side of Figure 1). The result of this social system analysis is used to give feedback to the system, and as such support institutional change. The central concept in IAD, is the 'action arena', in which individuals (or organizations) interact, exchange goods and services, engage in appropriation and provision activities, solve problems, or fight. The action arena is described by the participants (who have a set of resources, preferences, information, and selection criteria for action) and the action situation: the actual activity (or 'game') that is to be understood.



Figure 2.1 – The IAD framework (Ostrom et al., 1994).

The action arena influences and is influenced by the social system that it is embedded in. What happens in the action arena leads to patterns of interaction and outcomes that can be judged on the basis of evaluation criteria. The action arena itself is influenced by attributes of the physical world (e.g., climate, technological artefacts), the attributes of the community in which the actors/actions are embedded (e.g., demographics, culture), and the set of rules that guide and govern actor behaviour.

Although physical world and community affect the action arena, it is the rules of the game that actually define it. Therefore, in IAD quite some attention is given to rules. These rules can be analysed within three distinct layers: the operational, the collective choice and the constitutional choice layers. The different layers relate to different time-frames: day-to-day activities fall within the operational level, the collective choices determine what operational activities take place and these are reviewed over a 5-10 year time frame, whereas the constitutional level determines how the process of collective choice is organized (which is a long-term process). These rules are decomposed into a structure, referred to as the grammar of institutions, or ADICO (Crawford and Ostrom, 1995) which will be explained in detail and used throughout the chapters.

The IAD framework has been in development for more than 30 years using many case studies (e.g., (Yandle & Dewees 2003; Gordillo & Andersson 2004; Wynne 1989; Oakerson 1992)) where the concepts have proved to be robust. 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 in being used for agent-based model development.

#### Actor-centred Institutionalism

Actor-centred Institutionalism (ACI) is a framework that combines individualism with institutionalism to study governance and self-organization in social systems (Scharpf, 1997). Similar to Ostrom, Scharpf also believes that institutions are more tangible to study than internal actor behaviour. While IAD provides detailed description of institutions, ACI explains more about the actors; describing how institutions influence actors in terms of perception, preference, capability and interaction. The main aspects of the ACI framework are *actors*, *institutions* and *actor constellation*.

The first step in describing a social system within the ACI framework is to identify the actors whose choices will determine the outcome of the system. This is done by identifying the set of interactions that these actors are actually involved in. Actors in this framework are assumed to be partially rational in the sense that they will attempt to maximize their own self-interest. However, unlike other theories such a neoclassical economics (Jones, 1965), ACI does not assume complete information nor unlimited computational power for actors. This framework also assumes actor decision making on the basis of perceived reality rather than an objective reality similar to artificial agents. In addition, actors have certain capabilities and action orientations that depend on actor type.

There are two different types of actors characterized by their preferences, perceptions and capabilities: individual, composite. Composite actors are further divided into 'collective' and 'corporate' actors. While collective actors highly depend on and are guided by their members (e.g., a government), corporate actors are more independent in the sense that the members are supposed to have more neutral opinion (e.g., a school). On the other hand, while corporate actors like individual actors can make free choices, collective actors are not autonomous in that sense: their choices depend on the choice of the members (whether related to separate or shared goals) of the collective (Scharpf, 1997).

The second aspect of ACI are institutions. Similar to the IAD framework, ACI also restricts institutions to a system of rules (i.e., norms, regulatory rules, etc.). This way, institutions are not only perceived as a matter of evolutionary development but can also be intentionally created and changed by specific actors (Scharpf, 1997). Scharpf (1997) emphasizes that institutions enable and restrict actor behaviour but they do not fully determine behaviour, which leaves flexibility for strategic and tactical choices of actors. Nonetheless, to avoid models becoming overly complex by considering every institutional and non-institutional factor, ACI uses levels of abstraction starting from institutional explanations and if those are not sufficient, going into more actor-centric factors.

Finally, actor constellation is important when the strategies of different actors are interdependent. Actor constellation describes the actors involved, their strategy options, the outcomes associated with these options and their combination, and the preferences of the actors over the outcomes (in a game theoretic setting) (Scharpf, 1997).

#### 2.2.2 Theory of Structuration

The objective of the theory of structuration is to conceptualize human knowledgeability and its involvement in actions ordered across time and space, in order to understand and analyse social systems (Giddens, 1984). This theory complements the previous two frameworks in the sense that it has more details on actions and the structure of the social system rather than institutions (as in IAD) and actors (as in ACI), while still giving importance and value to the latter two aspects.

The main concepts in the structuration theory are agency, structure, system and duality of structure (Giddens, 1984):

**Agency** refers to the capability of people performing actions and *not* their intentions for performing those actions (Giddens, 1984). Nonetheless, agents have reasons for their actions and are able to elaborate discursively upon those reasons. The reasoning and rationalizing about actions is different from their motivation. While reasons refer to the grounds for action, desires refer to the wants. Actors can describe their intentions and reason about their actions but they cannot reflect on their motives.

**Structure** is defined through a set of properties that bind time and space to social systems making similar practices repeat leading to some 'systemic' pattern. In other words, structure refers to rules and resources independent of time and space distinguished by the 'absence of the subject'.

**System** on the other hand is the situated activities of actors through time and space given the structure. Giddens (1984) emphasizes that normative commitments (i.e., rules) are only one sector of the rationalization of action. Actors with their decision makings can create a variety of manipulative attitudes in the system.

**Duality of structure** refers to the fact that the rules and resources that result in social action are themselves means to perform actions. Therefore, structure is not a static concept within which we can study social behaviour but rather a dynamic and ongoing process (Giddens, 1984).

In essence, the idea is to find out how the concepts of action, meaning and subjectivity should be specified and how they might relate to notions of structure and constraint, summarized in the following quote:

The basic domain of ... the social sciences ... is neither the experience of the individual actor nor the existence of any form of societal totality, but social practices ordered across time and space. Human activities are recursive. ... They are not brought into being by social actors but continually recreated by them via the very means whereby they express themselves as actors. In and through their activities as agents, they reproduce the conditions that make their activities possible. (Giddens, 1984) To apply the structuration theory, social analysts study social activities under three different categories of situations: chain of events, complex relation among events and institutional practices. In the first case, a pattern of unintended consequence initiated by a single event results in a chain of events taking place one after another. In the second case, there is a complex relation between the activities of agents, leading to an emergent phenomenon that may not have a one to one relation with the consequence of single actions. This is also pointed out by game theorists with a different perspective: "the outcome of a series of rational actions, undertaken by individual actors, may be irrational for all of them" (Giddens, 1984). In the third category, the analyst is interested in understanding the mechanism for institutional practices. In such cases, the unintended consequence of actions forms the recognized condition for further action in a causal loop.

#### 2.2.3 Social Mechanisms

Social mechanism is an analytical approach to explain association between events (phenomena) in a social system (Hedstrom and Swedberg, 1998). Rather than looking at the system from the perspective of actions and structure (structuration theory), the social mechanism approach describes a system by explaining social phenomena and their relation. This approach is especially useful for our purpose because it provides the possibility for explaining emergent phenomena which is not quite feasible in the previous theories and frameworks even though it has been addressed in all.

Social mechanism is based on the macro-micro-macro model (Coleman, 1986). Hedstrom and Swedberg (1998) claim that every global event can only be explained if it is conceptualized in terms of macro-micro, micro-micro and micro-macro transitions. The first step to analyse a social system is to try to establish how macro-level phenomena affect the individuals. In the second step, the analyst studies how these individuals take those macro-level phenomena in. The final step is to find out how individuals through their actions and interactions generate macro-level phenomena.

The typology of social mechanism which uses the macro-micro-macro model is illustrated in Figure 2.2. The first type of mechanism, called situational mechanism explains the macro-micro relation. An individual is exposed to a social situation that affects him in some way. This mechanism links some social structure to the beliefs, desires and intentions of an individual. The second type called the action-formation mechanism shows the internal procedure of how individuals select an action to perform, based on their believes, desires, intentions and opportunities. The third type of mechanism which covers the micro-macro state is called transformational mechanism where individuals interact with each other and the environment transforming individual action to some kind of emergent collective outcome. The first two types of mechanism are internal to individual agents but the third type is external, involving a number of individuals.



Figure 2.2 – A typology of social mechanisms (Hedstrom and Swedberg, 1998).

#### 2.2.4 Reflection on the Social Foundation

This research will use the combination of the introduced theories and frameworks explained in this section to build a framework for conceptualizing ABMs with the goal of incorporating social structures into these models. The IAD framework has an overview of the whole system which is highly instrumental for giving a wider angle to the system which we want to make a simulation of. This framework provides a detailed structure for institutions (Crawford and Ostrom, 1995) which is our selected element for building social structure. The other aspects of the system, even though present in the framework, have less details.

The ACI framework (Scharpf, 1997), also builds on the institutional perspective but goes more in depth into who the actors are. However, it provides less detail about what they actually do in the system. The structuration theory not only goes into the details of actions and the required resources for performing those actions, but provides a definition for a system as a whole and addresses time and space (Giddens, 1984). Finally, while still keeping the individual and institutional perspective, the social mechanism theory looks at the system in terms of events. What this theory truly adds is the description for emergent phenomena in social system through individual and institutional behaviour and reasoning (Hedstrom and Swedberg, 1998).

#### 2.3 Computational Foundation

The combination of the theories and frameworks introduced in the previous section seems conceptually comprehensive and rich. However, using this combination to develop simulations also requires some foundation on the computational side. In this section, we will explain what ABMS is, in more detail, in order to show how it coincides with the social foundation. We will also explain the method we take to map the conceptual model of a socio-technical system into a software simulation.

#### 2.3.1 Agent-oriented Software

Agent-orientation is a relatively recent approach for developing systems in software engineering, artificial intelligence and simulation domains. An agent-oriented software is built as a collection of computational entities called agents that communicate among themselves and with their environment to achieve some goal. This environment can be virtual (i.e., a software), real or a combination.

In its simplest form, an agent is a software entity that is situated in an environment and is capable of autonomous action in order to meet its design objectives (Wooldridge and Jennings, 1995). With this simple definition, even an automatic light switch can be considered as an agent. However, a more complete definition is given to an *intelligent* agent who is in addition, reactive, proactive and has social abilities (Wooldridge and Jennings, 1995). Reactivity in agents means that they are able to perceive the environment and respond in a timely fashion to meet their design objectives. Proactivity means that agents can show goal-directed behaviour. Social ability means that the agents are capable of interacting with other agents and with humans in order to satisfy their objectives (Weiss, 1999).

Intelligent agents serve various purposes including distributed intelligent systems, intelligent control and social simulation. Those agent-oriented software that are developed to serve a purpose other than simulation are usually referred to as multi-agent systems (MAS) (*see* Chapter 1). If an agent-oriented approach is taken to develop a simulation, that approach is referred to as *agent-based modelling and simulation (ABMS)* and the software is called an *agent-based model (ABM)*. The main distinction between MAS and ABMS is that the goal of a MAS is the system itself but the goal of ABMS is understanding what comes out of the system (Luck et al., 2003). In addition, in ABMS, the agents represent real world components, while in MAS, agents are used for what they do (Luck et al., 2003). The focus of this research is on ABMS.

#### Agent-based Modelling and Simulation

ABMS is a relatively new approach for social simulation compared to other modelling and simulation approaches such as System Dynamics (Forrester, 1961) and Discrete Event Simulation (Banks et al., 2000). ABMs are powerful models that represent real-world systems with a pertinent degree of complexity and dynamics (Luck et al., 2003). Having identified a target phenomenon, ABMS is used to describe the system of which the phenomenon is a property or outcome of and finally evaluate the effect of individual behaviour and interaction on the target phenomenon (Conte et al., 2001).

Traditionally, the minimum requirements for building ABMs are: the consideration of the heterogeneity of agents across a population, the development a virtual environment that represents the social system, the identification of patterns of system behaviour that emerge from these agents' interactions (Macal and North, 2010), and data analysis to gain insight into the simulated system (Heath et al., 2009). The verification and validation of the model is also of crucial importance because the simulated system needs to represent some required aspects of the reality in order to be reliable for answering questions (Heath et al., 2009).
One broad application area of ABMS is modelling social system to support policy or managerial decisions by providing plausible explanations of observed phenomena (Luck et al., 2003; Moss, 2002).

To aid model development, some researchers propose guidelines on how to build agent-based models (e.g., (Drogoul et al. 2003; Heath et al. 2009; Gilbert & Troitz-sch 2005)). The general steps include conceptualisation, design, construction, and evaluation (Ramanath and Gilbert, 2004). Other researchers provide software platforms to ease and facilitate ABMS (e.g., Repast (North et al., 2006), Netlogo (Tisue, 2004), Swarm (Minar, 1996)).

Although there are various tools and methods for ABMS, building agent-based models is still not fully accessible to social scientists and policy makers who are not experienced in programming. Furthermore, considering various methods for participatory ABMS (e.g., (Ramanath and Gilbert, 2004), the current tools do not directly facilitate the participatory methods either.

As discussed in Chapter 1, social structures are not modelled in current ABMs. Therefore, we will be using theoretical descriptions of socio-technical systems to incorporate social structures into these models. We will be using MDSD to perform this process, while also considering the aforementioned practical drawbacks of ABMS.

#### 2.3.2 Model-driven Software Development

Each software system has an inner structure that directly influences the quality, performance, maintainability and portability of the software. The structure of a software however, is difficult to recognize at programming level because of the very low abstraction and amount of details (Stahl and Völter, 2006).

Model-driven Software Development (MDSD) provides a level of abstraction that gives an overview of the inner structure of the software and reduces the details required for development, to the essence (Stahl and Völter, 2006). In other words, by removing details that are irrelevant for a given viewpoint, MDSD helps us understand the core more easily. Figure 2.3, depicts the MDSM approach. To develop an executable software with this approach, a model is built using a high-level language called a meta-model. A software platform then takes some transformation rules to generate an executable software from the model.

**The Model** A model is an abstract representation of a system. A metamodel is a set of concepts and relations highlighting the common properties of a class of models. It is used to define the syntax of a modelling language at an abstract level (Kent, 2002). The relationship between a model and a meta-model is illustrated in Figure 2.3.

**The Transformation** Transformations are a set of protocols that use a model as input to produce a different type of model. The output model can be the actual executable model (i.e., in programming code).



Figure 2.3 - In MDSD, a system is represented by a model that conforms to a meta-model and is transformed to an executable model through a platform.

**The Platform** The platform can be as small as a piece of software code that translates a model of a system into some other model using the transformation protocols.

Following the MDSD approach has additional benefits for social simulation (Stahl and Völter, 2006):

- 1. It speeds up the development process of the simulation by giving more structure to the software, specifying the procedure of development and enabling automation (generating executable code).
- 2. It manages 'complexity through abstraction' because modelling languages enable 'programming' at a more abstract level.
- 3. It manages complexity through structure because modelling languages provide predefined place-holders for software components.
- 4. It separates tasks in the simulation development process: analysts perform system analysis, designers design the simulation, programmers code the system and debuggers evaluate and fix the simulated system.
- 5. It facilitates reuse and regeneration of simulation or its components because user knowledge becomes widely available in software format.
- 6. It improves software quality, performance, maintainability and portability because the architecture of the software may recur uniformly in the implementation.

By building a modelling language from the combination of social theories explained in Section 2.2, the MDSD approach facilitates model development from the conceptual analysis of a socio-technical system.

#### 2.3.3 Formal Languages

A meta-model provides a generic description of a set of similar models. For example, a model of a socio-technical system such as a bio-gas energy system, can conform to a meta-model that defines the concepts and relations in the class of socio-technical systems. Meta-models are used as modelling languages to conceptualize a system (Kent, 2002).

Meta-modelling is common practice in software engineering, particularly agentoriented software development (e.g., AGR (Ferber et al., 2005), INGENIAS (Pavon et al., 2005), TROPOS (Bresciani et al., 2004) and MOISE (Hannoun et al., 2000)). To develop a meta-model from our collection of social theories and frameworks (*see* Section 2.2), we learn from and adapt an existing language called OperA (Dignum, 2004). There are several reasons for choosing OperA among the existing metamodels. First, OperA is a meta-model for defining agent organizations in MAS. Therefore, institutions (referred to as norms in OperA) and social roles (i.e., positions in IAD) are already formulated. Second, compared to other meta-models, OperA covers more of the eight modelling dimensions as specified by (Coutinho et al., 2009), namely: structure, interaction, function, norm, environment, evolution, evaluation and ontology. Third, unlike most other meta-models that suffice to define the language, OperA follows MDSD to produce agent software and provides tool support for that purpose (OperettA (Dignum and Aldewereld, 2010)).

**OperA.** The OperA meta-model proposes an expressive way for defining open organizations distinguishing explicitly between the organizational aims and the agents who act in it. That is, OperA enables the specification of organizational structures, requirements and objectives, and at the same time allows participants to have the freedom to act according to their own capabilities and demands.

The OperA meta-model consists of three interrelated models illustrated in Figure 2.4. The **Organizational Model** (OM) specifies the *means* to achieve such objectives. That is, the OM describes the structure and global characteristics of a domain from an organizational perspective. Organizational objectives are achieved through the action of agents. The organizational model consists of four structures. The *social structure* of an organization describes the roles holding in the organization. The *interaction structure* describes a partial ordering of meaningful scene scripts. A *scene script* describes a scene by its players (roles), its desired results and the norms regulating the interaction. The aim of the *communicative structure* is to describe the communication primitives. Finally, the *normative structure* defines the norms that regulate roles, and that specify desired behaviour that agents should exhibit when playing the role.

The **Social Model** (SM) specifies how agents enact roles in an organization. Agent capabilities must be checked against role requirement and roles are assigned on the basis of this. In particular, it must be checked that the agents have the required capabilities. The resulting agreement is fixed in a social contract between agent and organization, against which the activity of the agent at runtime can be evaluated. Note that agents are still free to decide on compliance or violation of their social contracts. Different agent's 'personalities' will result in different role



Figure 2.4 – The OperA Overall Architecture

enactment behaviours, from social to fully egoistic.

Although OperA is a comprehensive meta-model, there are still many concepts that need to be added or revised in order to make it suitable for ABMS of sociotechnical systems. For example, agents and technological concepts are not defined in OperA. One other issue is that since this meta-model is primarily designed for MAS, the concepts are too technical for non-MAS experts. Thus, we will use OperA only where it complies with our research objectives.

## 2.4 Conclusion

In this chapter, we introduced the social and computational theories, frameworks and approaches that will form the fundamentals of this research.

We introduced the concept of institution which will be used as an analytical perspective. The IAD framework will be the basis for this research. However, we will take other theories into account when adding details to IAD, namely: ACI, structuration theory and social mechanism. Our selection of social theories and frameworks is done in a complimentary manner, trying to provide an overall comprehensive yet detailed explanation of socio-technical systems in order to incorporate social structures into ABMs.

On the computational side, we will develop agent-based simulations as they provide a natural representation of socio-technical systems. To bridge the gap between theoretical analysis and ABMS, we will use MDSD and build a meta-model that will then be used to develop simulations from high level concepts. We will propose this meta-model in the next chapter.

## Part II

# CONCEPTUALIZATION OF AGENT-BASED MODELS

There are three constants in life...change, choice and principles.

Stephen Covey



## 3.1 Introduction

To gain insight into socio-technical systems, social scientists and policy makers use various theories and frameworks to construct analytical models of the system (Hedstrom and Swedberg, 1998). In Chapter 1, we explained why agent-based models can be used to gain insights into socio-technical system and how they can be improved by incorporating social structures into the models. We selected institutional analysis as a perspective and introduced a selection of analytical theories and frameworks that can in combination describe socio-technical systems in order to conceptualize agent-based models.

Conceptual modelling is common practice in software engineering. Conceptualization in this field, which entails describing the set of concepts that will constitute the "building blocks" of the software, leads developers to better capture, analyse and understand what they are actually programming (Winograd et al., 1996; Nikolic and Ghorbani, 2011). As explained in Chapter 2, a formal description of this set of concepts that is used to describe a model and its properties is called a meta-model (Atkinson and Kuhne, 2003). In software engineering, meta-models are widely used. On the one hand, a meta-model provides the conceptual richness required 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.

Meta-models are gaining recognition in ABMS. ABMS platforms such as Repast (North et al., 2006), Netlogo (Tisue, 2004) and Ascape (Parker, 2001), have in fact seen the necessity of meta-models for their platforms and are beginning to reverse engineer meta-models that would fit their platforms (Janssen et al., 2008). Other researchers advocate using meta-models when constructing agent-based models of

social systems (e.g., (Hassan et al., 2009; Janssen et al., 2008; Sansores and Pavón, 2005)). However, this practice is still far from mainstream and even those researchers that do use meta-models, have not invested in the conceptual comprehensiveness of their meta-models.

For example, Hassan et al. (2009); Sansores and Pavón (2005); Garro and Russo (2010), propose meta-models for their ABMS tools. Their meta-models however, do not include social structures such as norms and culture which are major building blocks of a social system (Ostrom, 2005; Scharpf, 1997) and one focus of this research. Furthermore, their agent concept lacks features such as personal values and preferences that affect agent behaviour. As another example, Iba et al. (2004), only define basic ABM concepts (e.g., agent, relation and entity) in their models.

The goal of this chapter is to propose an ABMS meta-model for conceptualizing socio-technical systems. By developing this meta-model, not only we incorporate social structures into ABMs, we seek two additional benefits: (1) to bring ABMS within the reach of policy analysts and social scientists, especially those who are less familiar with programming or modelling, (2) to enable participatory model development in order to include more problem domain insights into models and increase the reliability of simulation outcomes.

In this chapter<sup>1</sup>, we first describe a socio-technical system with the theories and frameworks introduced in Chapter 2, Section 2.2, to show how they in combination provide an abstract overview of a system. In Section 3.3, we propose a meta-model called MAIA<sup>2</sup> (Modelling Agent system based on Institutional Analysis) that is based on the system description. MAIA formalizes the components of the IAD framework and extends it with other theories and frameworks to provide a descriptive modelling language for developing ABMs. In Section 3.4, we explain how to use MAIA and how it supports participatory model development. In Section 3.5, we define a set of criteria for evaluating MAIA. We partially evaluate this meta-model in Section 3.6 and Section 3.7, and conclude in Section 3.8.

## 3.2 Describing a Typical Socio-technical System

A socio-technical system is bounded in time and space and shaped by social *structure* (Giddens, 1984). It consists of many *actors* who perform actions and interact with each other in what is called the *action arena* (Ostrom et al., 1994):

#### Structure

The structure of the system is both the means to organize the system as well as the outcome of that system (Giddens and Turner, 1988). It consists of *resources* and *institutions*. Besides natural components, resources include the technical aspects of the system and information (e.g., price of gold). Institutions are the more enduring

<sup>&</sup>lt;sup>1</sup>This Chapter is partly published in (Ghorbani, Dignum and Dijkema, 2012; Ghorbani, Bots, Dignum and Dijkema, 2013).

 $<sup>^{2}</sup>$ 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 computational models while frameworks can be used for various purposes.

features of the socio-technical system (Giddens, 1984). They influence the payoffs for performing actions which in turn influence actors' decision making and behaviour (Scharpf, 1997). If institutions are in the form of sanctioned rules they reduce the range of potential actions by actors. However, institutions are not deterministic because actors are intelligent entities. They have considerable scope for strategic and tactical choices, and can make decisions that sometimes bring them to violate norms and rules (Giddens, 1984; Scharpf, 1997).

#### Actors

Besides the institutional context, the behaviour of actors depends on their perception, preferences (based on their subjective needs and evaluations), personal values, resources and capabilities (Scharpf, 1997; Hedstrom and Swedberg, 1998). Actors may also consider their objective needs and the payoffs received by other actors (Scharpf, 1997). Nonetheless, actors have limited information and computational power which means that they will not always be able to make the most profit maximizing decisions (Scharpf, 1997).

Actors make decisions about those actions that they are in the *position* of performing. In other words, actors take positions or roles in different situations to perform actions (Ostrom et al., 1994). These positions determine their institutional responsibilities and competencies with assigned resources (Scharpf, 1997). The actions associated to roles are impossible to perform without institutionalized rules governing them (Scharpf, 1997).

#### Action Arena

The action arena comprises and is formed by (inter)acting actors in a set of action situations (Ostrom et al., 1994). 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), information related to the situation, expected outcomes of the situation, costs and benefits and the actions actors perform.

In most activities, the scope of control is limited to the immediate contexts of action or interaction. Social commitments (i.e., institutions) set the necessary preconditions for performing actions (Scharpf, 1997). However, other preconditions must also hold for actors to perform actions (e.g., physical conditions) (Giddens, 1984). The reproduced conditions of one action may also become unacknowledged preconditions for consecutive actions (Giddens, 1984). These conditions may be the unintended consequences of the previous action or intentional. To explain actions, the reflexive part of an action is distinguished from its motivations (Giddens, 1984).

Finally, what happens in the action arena of the system leads to patterns of interaction and outcomes that are judged on the basis of evaluative criteria that are defined by the analyst (Ostrom et al., 1994).

The above description is abstract enough to explain any kind of socio-technical system. To summarise, it explains that the structure of a system influences and partly determines actor decision making which in turn results in emergent patterns of interaction and outcomes. This is the macro-micro-macro model (referred to as the bathtub model) social scientists use to analyse systems ((Hedstrom and Swedberg, 1998; Coleman, 1986)). We use this general view to develop ABMs and propose the following proposition:

**Proposition 1** In a social simulation, agents take structure as input and interpret. They will make decisions to execute actions which will lead to patterns of interaction and emergent outcomes that may even lead to change in the primary structures.

This proposition is different from the common approach in ABMS which is currently viewed as a micro-macro modelling approach in what Epstein (2006) calls the generative social science. We use macro structures in ABMs (i.e., institutional structure, physical structure) to study the influence of structure on agents' decision making and behaviour. The challenge is to propose a systematic way to integrate these structures into models that are built from bottom-up and that emerge from individual action and interaction.

In order to address this challenge and formalize the concepts described above into the MAIA meta-model, we elaborate on the following issues with a computer science perspective:

- 1. We take the definition of actors, in the aforementioned social theories, for computational agents.
- 2. We classify the concepts into various categories to separate agents from macro structures. This also results in a more understandable software architecture.
- 3. We formalize the relationship between various categories and concepts, to elaborate on the link between agents and macro level system structure, grounding the overall behaviour of the system.
- 4. We provide further detail on the characterization of concepts and system components in terms of their definition and attributes.

The formalization of the above description of a socio-technical system and the results of our elaboration are presented as the MAIA (Modelling Agent systems based on Institutional Analysis) meta-model which will be described in the next section.

## 3.3 The MAIA Meta-model

In this section we present the concepts and relations that constitute the MAIA meta-model<sup>3</sup> with the help of an illustrative example taken from one of the four case studies in which we have applied MAIA: the informal backyard recycling system in Bangalore, India.

<sup>&</sup>lt;sup>3</sup>More information about MAIA and the web-tool can be found at: www.maia.tudelft.nl.

**Case Description** 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 rules, norms, and shared strategies among the agents, which are not easy to understand or capture. Child labour and unsafe extraction (causing health and environmental hazards) are the major growing problems of this sector.

As a solution, the Indian government wants to introduce professional recycling companies that would take over the precious metal extraction. They would take the dismantled parts from the backyard recyclers, extract gold, and return money equivalent to the value of gold to the backyard recyclers. Data shows that the income received from professional companies is higher than the value of the inefficiently extracted gold. However, if backyard recyclers employ children and use unsafe chemicals, the income from the professional companies is lower than the value of the extracted gold. Therefore, this policy has not been successful.

One of the major questions that we aim to answer with this model is: 'How would rules for fining recyclers for child labour and harmful extraction, influence the economic situation of the recycling units and increase their incentive to work with professional refiners?' In the following sections, we use a simplified version of this case study to explain how an agent-based model of a socio-technical system can be conceptualized with MAIA.

Following the IAD framework, we organize the MAIA meta-model into five structures that serve as place holders (i.e., categories) for related concepts:

- 1. Collective Structure: agents 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.

During the conceptualization process, these structures are gradually filled by the modeller with information from the system being modelled. In the following description of MAIA, the meta-model concepts are written in *italic* font when first introduced, and case-specific concepts are in **typewriter** font.

#### 3.3.1 Agents and the Collective and Constitutional Aspects

The e-waste recycling system is viewed as a socio-technical system with several types of actors which we call *agents* in MAIA. Agents can represent individual as well as composite actors (cf. (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 and a professional recycling company are composite agents.

Independent of the roles they may assume in the society, agents have properties, personal values, 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<sup>4</sup> as their personal values. 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 (cf. (Giddens, 1984)). For example, all workers lose energy when they work.

Agents make decisions about the tasks they perform (cf. (Hedström and Swedberg, 1996)). For making decisions, agents do not need to have complete information about the situation nor are they necessarily rational (cf. (Scharpf, 1997)). To reflect these, the irrationality of decision making can be modelled as random choice, and the information which the agent uses in the decision process is limited. We assume that each decision making behaviour requires a criterion.

In sum, the Collective Structure specifies the attributes of all agents in the model (Figure 3.1).



Figure 3.1 – The Collective Structure in the MAIA meta-model.

To take part in the society, agents enact *roles* (cf. 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., 2005). A worker can take the role of a segregator who dismantles computers, the role of a refurbisher who refurbishes computer parts, or the role of an extractor extracting gold. He may also take the role of a unit boss and hire other workers.

Actors may take a role in the society, only if an *entry condition* is met (cf. (Ostrom, 2005)). 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,

 $<sup>^{4}</sup>$ This personal value may bring less wealth to them since they would use more expensive materials for gold extraction to keep safe.

can also take the role of a unit boss at the same time, provided that the conditions are met.

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).

When agents take roles, certain responsibilities become available to them (Scharpf, 1997; Giddens, 1984). 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.

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 and Ostrom, 1995). As we will describe in full detail in Chapter 6, the acronym ADICO refers to the five elements that an institutional statement can comprise: Attributes (the designated roles), **D**eontic (prohibition, obligation, permission), aIm, Condition (for the institution to hold), and '**O**r else'.

The 'aim' of a statement is an action (or a set of actions) taken by agents. The 'or else' specifies the *unique* and *explicit* sanction that applies if an agent does not comply with an institution. The 'or else' is itself an institutional statement.

Crawford and Ostrom (1995) show that institutional statements can be categorized into three types: *rules, norms* and *shared strategies.* Table 3.1 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.

Taking a role in the system does not force agents to follow the rules associated with that role (cf. (Giddens, 1984)). 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 aIm part of the ADICO statement) that is part of a rule, a decision making process occurs in which the agent decides whether to comply with the institution (cf. (Hedstrom and Swedberg, 1998)). For example, even though hiring children may result in fines, a unit boss may still decide to take this risk and hire children to

	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.

1 D'A 10 T	, C	· 1		· 1		1
Table $3.1 - Different$	types or	institutional	statements	ın tn	e e-waste	example.

save money. Then, when paying the child, he knows that it is common practice to pay half salary, but he may do otherwise if he evaluates that paying more or even less has a higher payoff for him. In his decision making, he may consider the poverty of the child for example. In that case, he is also considering the child's payoff (cf. (Scharpf, 1997)).

Figure 3.2 summarises the concepts we define in the Constitutional Structure of MAIA to capture the social context of a socio-technical system.

	Constitutional Structure													
Role					Institution						Dependency			
Name	Objective	Information	Institutional Responsibility	Institution	Entry Condition	Name	Attribute	Deontic Type	Aim	Condition	Or else	Depender	Dependee	Objective

Figure 3.2 – The Constitutional Structure in the MAIA meta-model.

#### 3.3.2 Agents and their Physical Context

Besides the agents defined in the Collective Structure and the social aspects defined in the Constitutional Structure, there are many *physical components* that need to be conceptualized in the model (cf. (Giddens, 1984; Ostrom et al., 1994)). Computers, refurbished parts, motherboards, gold, tools and waste are components that seem relevant for this model. Computers and refurbished parts, gold and waste all have weight and price/kg as their *properties*. These components have four *affordances* (i.e., what can be done with them), namely: be processed, produced, *bought* and *sold* by the agents. For example, an extractor would process motherboards to extract gold. Physical components can be accessed/used only by agents having a behaviour (or responsibility) associated with the affordances of the component. Besides properties and affordance, physical components may also have *behaviours* (e.g., **ageing** of a computer). Physical components also have *types*. A physical component may be *public* for every agent to use or *private* (i.e., restricted). All the physical components in the e-waste example (computers, gold, etc...) are private, but a road would be an example of an public physical component.

Specifying *composition* relations between the physical components may also be relevant for the e-waste model. In our model, a **segregated** computer consists of, on average, 3 kg of **refurbished parts**, 1 kg of **waste** and 0.5 kg of **motherboards**. Since collecting **gold** is the aim of processing motherboards, this composition is also important in the simulation. Although not relevant for this particular case, the *connection* between the physical components may also be specified. When implementing a spatial model, these connections show which physical components are attached to each other (e.g., a road network). Figure 3.3 summarises the concepts in the Physical Structure.



Figure 3.3 – The Physical Structure in the MAIA meta-model.

#### 3.3.3 Agents and their Operational Environment

The Operational Structure (Figure 3.4) describes the dynamics of the agents' influence on the system's state in relation to time and space (cf. system in (Giddens, 1984)). It defines the actions that the entities (agents and physical components) of the system are responsible for, and the (partial) order in which these actions are performed (cf. (Giddens, 1984)). 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 or unit bosses who have hired extractors, can decide to use unsafe chemicals for extraction, which will then decrease the energy level of the extractors.
- 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 perform 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.

Each of the entity actions (e.g., buy old computers) mentioned in the action situations has a *precondition*, which tests the actual feasibility of performing that action (cf. (Giddens, 1984), 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 (cf. (Hedstrom and Swedberg, 1998)). 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 he is a child, based on his personal values and other factors that influence his decision.

The order of entity actions in an action situation is specified by a plan (cf. (Giddens, 1984; Ostrom, 2005; Dignum, 2004)). Plans are defined recursively using these four types:

- 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. *choice*: 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).

#### 3.3.4 Answering Questions Using an Agent-based Model

Like any other software system, errors in simulations should be detected as early as possible starting from the analysis and conceptualization phase (Balci, 1997). There may be notable differences between simulation outcomes and reality which may be due to software bugs but also conceptualization errors (Ramanath and Gilbert, 2004). The Evaluative Structure (Figure 3.5) is inspired by the right-hand side

Operational Structure											
Entity Action	Role Enactment			Plan			Action Situation			Action Arena	
Action Body Performer Pre-condition Post-condition Post-condition ConditionNotDo Decision making		Agent Role	Entity Action	Name	Type	Plan	Condition	Name	Plan	Institution	Action Situtation

Figure 3.4 – The Operational Structure in the MAIA meta-model.

of the IAD framework (Ostrom et al., 1994) depicted in Figure 2.1. It provides 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?).

To make sure that the model implementation does not violate real-world conditions, constraints can be specified for variables during conceptualization. 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 action and a variable will help focus the analysis of the data after a simulation run.

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 units during the simulation run.

The calculation concept defines how each validation and problem domain variable is calculated. To analyse data and facilitate the visualization of results, we define independent variables. For example, for monitoring the **recycling unit size**, we use the independent variable **time step**.

	Evaluative Structure								
Proble	Problem Domain Variables Validation Variables								
Entity Action	Independent Variable	Dependent Variable	Calculation	Entity Action	Independent Variable	Dependent Variable	Calculation		

Figure 3.5 – The Evaluative Structure in the MAIA meta-model.

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 socio-technical systems. Figure 3.6 presents a complete overview of the MAIA meta-model in the form of a UML class diagram.

## 3.4 Using MAIA

The MAIA meta-model is a structured description of a socio-technical system. It helps the modellers to decompose and conceptualize a system into a set of concepts and relations that are the main building blocks of the socio-technical system they want to model. The conceptual model of a socio-technical system (e.g., e-waste recycling sector), referred to as the MAIA model, brings the theoretical analysis of socio-technical systems one step closer to simulation because the concepts and relations are more formally structured.

A MAIA model is presented in ten tables, three diagrams and two matrices<sup>5</sup> which are completed by the modeller for a specific socio-technical system. This model can be used by the modeller herself (if she knows programming) or any other programmer to build an executable simulation.

To increase the usability of the model and to ensure that the conceptualized model conforms to the meta-model, a software application called the MAIA tool has been developed (Ghorbani, 2012).

<sup>&</sup>lt;sup>5</sup> Tables: Agent, Physical Component, Role, Institution, Entity Action, Action Situation, Plan, Action Arena, Validation, Problem domain, *Graphs*: Physical Connection, Physical Composition, Role Dependency, *Matrices*: Scope Matrix, Reality Closeness Matrix





COLLECTIVE STRUCTURE Agents	O Add agent	Search	Collective Structure The Collective Structure specifies the attributes of all agents in the model
CONSTITUTIONAL STRUCTURE			
Roles	Label	worker	An agent (agent type) is an institution-driven
Institutions			entity in the system who must take roles to be
Dependencies	Properties	* age (number) * gender (number) * money (number)	able to perform tasks in a given system.
DUVSICAL STDUCTUDE		* familyID (number) * skillLevel (number)	Properties: The relevant attributes (e.g.
Physical components			entity represented as an agent
Physical connections	Personal values	× health () × safety ()	Personal Values: The intentions of an
Physical composition			agent (e.g. safety, happiness) which
OPERATIONAL STRUCTURE	Information	goldPrice, oldComputerPrice	Information: The information available to
Entity actions			an agent (e.g., market price for gold,
Plans	Physical	× oldComputers	the beliefs of an agent in the BDI
Action situations	componenta		architecture.
Role enactments	Possible roles	* extractor * refurbisher * unitBoss	Physical components: Similar to
Action arena			physical components (e.g. car, computer)
EVALUATIVE STRUCTURE	Intrinsic capabilities	* loseEnergy	Possible Roles: The roles that the agent     bas the potential of taking. Agents take
Problem domain evaluation			roles to act in the system.
Problem domain matrix	Decision making	× becomeUnitBoss, × hire × useDangerousChemicals	Intrinsic Behaviours: The behaviours an
Validation	criteria		agent has independent of the role he is taking (sleeping, talking). All the actions
Validation matrix		·	taking place in the system should be
	Î	Save O Cancel	reflected as an intrinsic capability of an
			role or a behaviour of a physical
	Add agent		component. • Decision Making Criteria: The criteria that
			the agent uses to decide to perform an action. Decisions are related to entity
			actions

Figure 3.7 – A worker agent card is being created in the MAIA web-based application.

#### 3.4.1 The MAIA Tool

The MAIA tool<sup>6</sup> (which can be opened in any web browser), guides the modeller throughout the conceptualization process by presenting the concepts in 'cards' and diagrams in the order they need to be filled in. The tool checks the input information for consistency and compliance with the meta-model. Besides facilitating communication and sharing of information, the MAIA tool also stores a model in XML (Extensible Mark-up Language) format. We will use XML for automatic generation of code in later chapters.

The MAIA tool is organized in five parts: one for each structure in the metamodel. Each part contains cards and diagrams for the concepts of that structure.

Conceptualization starts by identifying the agents and their attributes, and defining a card for each agent type. Figure 3.7 shows a worker agent being defined using the MAIA tool. Roles and physical components for this agent type are also defined as cards in the related pages.

After defining the agent types, the modeller can proceed by defining the roles,

<sup>&</sup>lt;sup>6</sup>The MAIA tool is a web-based application developed in Javascript, HTML and CSS. Several libraries were used, including Angular.js (a model-view-controller framework), Twitter Bootstrap (styling), and RaphalJS (for rendering graphs). We also used the Select2 and Angular-UI libraries, which both add additional widgets to the default set of controls that are offered by the Web Browser. The application needs no custom-built server backend; user models are stored directly in the user's Google Drive account, using the Google Drive API.

Label	hireRule	
Attributes	unitBoss	
Deontic type	may not	
Aim	hireWorkers	
Condition	worker.age < 12	
Or else	fineRule	
Label	paymentNorm	
Attributes	unitBoss	
Deontic type	must	
Aim	paySalary 50	
Condition	per day	

Figure 3.8 – Two institution cards for the e-waste system in the MAIA tool.



Figure 3.9 – Objective dependencies between roles in the e-waste example.

Add entity_action	Search	
Label	hireEmployee	
Performer	unitBoss	
Action body	hire	
Decision making	hireEmployee	
Precondition	have money	
Postcondition (do)	employee add to group, worker.employed = true	
Postcondition (not do)		

Figure 3.10 – The hire entity action card for the e-waste example in the MAIA tool.

Action situation	E-waste Market
Action situation	Production
Action situation	Employment
Action situation	SafetyInspection

Figure 3.11 – The action arena for the e-waste example in the MAIA tool.

institutions and role dependencies, in the constitutional structure section. Figure 3.8 shows two institution cards.

In the dependency graph partly depicted in Figure 3.9, the software automatically provides the nodes. The modeller connects the different roles (nodes) and labels the connections based on the objectives of the depender 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.

The physical structure section of the MAIA tool allows the modeller to define physical component cards similar to agents and roles. Furthermore, it specifies 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 3.9).

In the operational structure section, the modeller makes use of the already defined cards in other sections to specify the dynamic structure of the model. Figure 3.10 shows the hireEmployee entity action card. For this entity action, the unitBoss as the performer of the action, makes a decision about hiring employees based on his needs. The entity actions are put in plans and those are then organized in the action situation page to shape the actual action arena of the model presented in Figure 3.11. The action arena page orders and displays the action situations to show the general sequence of events taking place.

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 finally links the conceptual model to the expected outcomes of the agent-based model in the evaluative structure section, which contains: the scope matrix<sup>7</sup> (Figure 3.12), the reality closeness matrix, the validation table and the problem domain table. In each row of the scope matrix, the modeller defines the problem domain variable, the entity action related to the variable, and the type of relation between the problem domain variable and the entity action. The reality closeness matrix is similar to the scope matrix. The dependent variables, the independent variables and the method that is used to calculate the dependent variables are all defined in the two tables in this section.

The information captured in the evaluative structure is used for the analysis of 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.

	hireEmployee	hireChildren	produceWaste	produceRefurbishableParts
number of units	indirect •	none 🔻	none 🔻	none 🔻
number of workers per unit	direct *	direct v	none 🔻	none v
amount of recyled computers	indirect 💌	indirect 💌	indirect 💌	direct

Figure 3.12 – Part of the validation matrix for the e-waste example in the MAIA tool.

During conceptualization, the modeller is continuously stepping backward and forward through the pages 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, and/or export the data into an XML file.

#### 3.4.2 Programming a MAIA Model

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 briefly describe one way of transforming a socio-technical system described in MAIA (a MAIA model) to an object-oriented simulation program.

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.

 $<sup>^7\</sup>mathrm{This}$  matrix originates from the definition of scope rules by Ostrom (2005) linking actions to outcomes.

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 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)
FOR
END FOR
Analyze_data
END Simulation
```

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

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 which is determined by the modeller. Since the order of actions influence outcomes, this modelling choice should be taken with careful consideration. The 'lowest level' statements in Listing 2 (represented as procedures with an agent as the input parameter) denote atomic plans.

Time steps in the simulation are defined in relation to the actions the agents perform. In other words, we assume that each agent only performs a maximum of one action per *time step*. This is to enable the tracking of every decision and action and be able to study the consequence of each separately. The agent selects the action to perform, by checking the entity actions one by one. As soon as an entity action is executed (whether successful or not) the agent exits from the action arena, keeping track of the *plan* he 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.

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 perform 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 he 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. He 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.

1	BEGIN ActionArena(agent)
3	//market action situation
	//plan type: sequence
5	buy_old_Computers(agent)
	sell_products(agent)
7	
	//production action situation
9	//plan type: sequence
	dismantle_computers(agent)
11	make_refurbished_parts(agent)
	//plan type: alternative
13	$extract_gold_safely(agent)$
	$extract\_gold\_unsafely(agent)$
15	
	//employment action situation
17	//plan type: sequence
	$form\_recycing\_unit(agent)$
19	$calculate\_hiring\_need(agent)$
	// plan type: loop
21	WHILE(employee_not_found & workers $> 0$ )
	$\operatorname{findEmployee}(\operatorname{agent})$
23	END WHILE
	//plan type: sequence
25	$pay\_employees(agent)$
	$fire\_employees(agent)$
27	
	//safety inspection action situation
29	//plan type: sequence
	find_unsafe_extraction(agent)
31	find_child_employement(agent)
	//plan type: alternative
33	fine_violators_bribed(agent)
	fine_violators_not_bribed(agent)
35	END ActionArena

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

In each *entity action*, the preconditions are checked to see 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). 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. More details about the whole procedure will be the subject of Part II of this thesis.

#### 3.4.3 Participatory Model Development

One of the objectives of this research is to facilitate participation of various parties such as domain experts, policy analysts, social scientists and programmers in building agent-based models. These parties work together as a team towards shared objectives, with each contributing their own expertise (Gilbert et al., 2002). This has several benefits including the ability to integrate expert knowledge into the model, the possibility to increase trust in and give legitimacy to models and the opportunity to have professional simulations if programmers are also involved.

In most participatory model development processes, domain experts and problem owners are involved at two discrete intervals during the simulation project: early interviews for model specification and final communication of results. To increase the effectiveness of participation, these steps are repeated similar to the Extreme Programming methodology in software engineering (Ramanath and Gilbert, 2004). In most cases, initial interviews are taken to specify and design the model and during repetitions, card games and prototypes of the simulations among other methods are used to verify the model with stakeholders (Ramanath and Gilbert, 2004).

One other frequent method of participation is combining games with simulations (e.g., (Chu et al., 2012; Guyot and Honiden, 2006)). This way the stakeholders are put 'in the model', replacing artificial agents so that they can contribute their own tacit understandings directly, and at the same time gain a thorough understanding of what it would be like to select actions (Barreteau et al., 2001). This approach to participation can act as an additional source of knowledge about the system being modelled (Gilbert et al., 2002).

MAIA acts as a tool to enable participatory model development and facilitates the aforementioned participatory methods. It can be used as a means of communication with domain experts and problem owners for model specification. A MAIA model can be used as a source of documentation during interviews to structure and decompose the information problem owners and domain experts provide. Furthermore, these parties can themselves use the web tool to provide information for the modellers and therefore collectively build a model. The key element here is that the concepts in MAIA are kept at a generic level so that they are understandable to a wide range of users.

Participation is not limited to domain experts. For larger and more complex agent-based models, model development can be performed by a team of experts including analysts, designers and programmers in addition to the problem owners and domain experts. MAIA can also be used as a source of communication between analysts (or modellers) who build the conceptual model and programmers who make the simulation.

Figure 3.13, shows the roles that can be involved in a participatory ABMS process using MAIA. As the figure illustrates, various parties who may have little familiarity with computational tools can take part in building a conceptual model. A MAIA model can be developed through various processes from interviews and workshops to prototype specifications. On the other hand, a conceptual MAIA model can be used by programmers or software designers to build executable simulations which may be prototypes or the final simulation. The process can be repeated as many times as required following the state-of-art methods in participatory model development. One important benefit of MAIA in this process is the separation of tasks between experienced and non-experienced programmers.



Figure 3.13 – Participatory ABMS and separation of tasks using MAIA.

## 3.5 Evaluating the MAIA Meta-model

Many software practices, tools and methods are developed every year. However, only few of them are evaluated. Nonetheless, evaluation still remains a common issue that researchers face when developing tools and methods.

The literatures on software evaluation is diverse. In this area, great attention is given to the evaluation of software products (e.g., (Lawis et al., 2001)) and to some extent, conceptual models (e.g., (Moody, 2005)). However, there are a limited number of articles on the evaluation of the actual tools and meta-models that *make* those products and conceptual models. Even so, those research that do evaluate methods and tools focus on *specific purpose software* and/or *specific evaluation dimensions*. For example, Koua et al. (2006) provide guidelines for evaluating the usability of geo-visualisation environments, Desurvire et al. (2004) evaluate the playability of software games, and Reiser and Dick (1990) focus on instructional software. Likewise, the generic software evaluation methods presented in the literature address particular evaluation dimensions such as usability in (Kitchenham et al., 1997; Segars and Grover, 1993) and effectiveness in (Card et al., 1987).

The evaluation dimensions and methods are so diverse in the literature that we can only define them in relation to the specific purpose of the software tool or method. After clarifying the purpose and defining the evaluation criteria, we would then be able to select and use existing evaluation methods. These methods include field experiments, surveys, case studies and comparison with similar tools (Kitchenham et al., 1997; Moody, 2005).

The goal of this section is to propose and define evaluation criteria for the MAIA *meta-model* and provide guidelines for the evaluation procedure. We define the evaluation dimensions in relation to the objectives of MAIA as an agent-based modelling framework. These are to (1) integrate social structures into agent-based models to facilitate the development of more comprehensive simulations and to (2) bring ABMS within the reach of more social scientists and enable participation of various parties.

#### 3.5.1 Defining the Evaluation Criteria

Considering our objectives, we identify four criteria to evaluate MAIA. To find out whether MAIA leads to more comprehensive agent-based models we have to find out whether the concepts in MAIA are *complete* for defining the various aspects of a socio-technical system especially social structures. Furthermore, we need to find out whether the concepts and relations in this framework are *sound*. In other words, we determine whether they are consistent in terms of definitions and relations, if there are any cyclic definitions in the meta-model and if all the concepts and relations are used (van der Aalst, 1997; Dehnert and Rittgen, 2001).

In addition, we have to investigate whether MAIA actually brings ABMS within the reach of the inexperienced users. Therefore, MAIA should be *useful* to the users and *usable* for ABMS. MAIA should also be *parsimonious*; it should be simple and have minimum yet sufficient number of concepts and relations (Sivo and Willson, 1998).

#### **Conceptual Soundness**

Soundness is a term often used for the evaluation of logical systems such as programming languages (Drossopoulou et al., 2000; Wright et al., 1994). Conceptually, the *soundness* of a meta-model like MAIA means that there are clear associations between the concepts in MAIA, and that all the model concepts and relations are used during simulation development. In other words, a conceptualized MAIA model is translatable into a simulation.

#### Completeness

Completeness is also used in the evaluation of logical systems. In general, the completeness of a meta-model implies that it represents every aspect of the system it is modelling. To narrow down the scope of this evaluation criterion and make it more tangible, we define completeness in relation to IAD because it is the basis of MAIA. IAD defines institutions as the building elements of social structure and specifies its relation with other components of the system. Therefore, compliance with IAD confirms that we have incorporated social structures into ABMs and considered their relation with other aspects of a system. Comparing MAIA to the IAD, benefits the evaluation of MAIA for two reasons. First, IAD is a mature framework that has been in development for over 30 years and used in many case studies in over 21 countries (Ostrom et al., 1994; Ostrom, 1999). Second, IAD has already been assessed in (Ostrom, 1999).

#### Parsimony

Parsimony<sup>8</sup> is a standard way for evaluating models (Sivo and Willson, 1998). It refers to the idea that under the same circumstances, a model with fewer parameters is better than a more complex model (Marsh and Hau, 1996). Parsimony in our context means being able to conceptualize a socio-technical system with minimum redundancy. To find out how parsimonious MAIA is, we have to identify the conceptually redundant variables which would make no difference in the amount of information captured by the model, if omitted from the framework.

#### Usability and Usefulness

Usability and usefulness are common features for evaluating tools and methods in computer science (Segars and Grover, 1993). In general, these two concepts refer to 'the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context' (Benyon et al., 2005). Usability concerns the ease-of-use and ease-of-learning, and usefulness addresses the effectiveness of a tool or method in serving its purpose (Segars and Grover, 1993). Therefore, while usability relates to the tool itself, usefulness is more about the content of the tool (Tsakonas and Papatheodorou, 2006).

Usefulness and usability of software tools or methods are defined through user experience (Segars and Grover, 1993; Sears and Jacko, 2007). Therefore, since the objective of the MAIA meta-model is to support inexperienced modellers, we believe this evaluation dimension of MAIA to be the most important aspect that requires extra attention.

Segars and Grover (1993), propose several features that define the usefulness and ease-of-use (i.e., usability) of a tool or method in Information Technology. Usefulness indicators include effectiveness, job performance and increase in productivity. The usability indicators are easy to use, easy to learn, easy to become skilful and clear/understandable. For information services and digital libraries, Tsakonas and Papatheodorou (2006) and Tsakonas and Papatheodorou (2008) respectively define usefulness indicators as relevance, reliability, level of information provided and timeliness (i.e., how current the information resource is). Their usability features however, are very similar to (Segars and Grover, 1993), addressing ease of use and learnability.

Usability indicators of a tool are defined similarly in various research (e.g., (Segars and Grover, 1993; Tsakonas and Papatheodorou, 2006; Lam et al., 2009; Tsakonas and Papatheodorou, 2008; Thomas and Fischer, 1996)). However, since usefulness is related to the content of a tool, we must take the purpose of MAIA into account.

<sup>&</sup>lt;sup>8</sup>The law of parsimony is Occam's razor principle, stating that unless a theory provides more powerful explanations, it is always better to select simpler theories with lowest number of assumptions (Baker, 2011). The simplest available theory does not need to be the most accurate.

### 3.5.2 Validating the Completeness and Verifying the Soundness

Conceptual evaluation of MAIA involves validating its completeness and verifying its soundness:

**Completeness.** To find out how complete MAIA represents IAD, we will compare it to this framework. IAD involves various elements (Smajgl et al., 2009): definition of institutions (Ostrom et al., 1994), typology of rules (Ostrom et al., 1994), the multi-level framework (Kiser and Ostrom, 2000), the ADICO statements (Crawford and Ostrom, 1995) and the design principles (Smajgl et al., 2009). Since MAIA is built on the 'conceptual model' of IAD (Chapter 2, Section 2.2.1) and ADICO statements, we will only compare these two elements of IAD with MAIA.

**Soundness.** To verify the soundness of MAIA, we will use UML class diagrams as suggested in (Evans, 1998) with additional well-formed rules given in Object Constraint Language (OCL) (Evans et al., 1999; Czarnecki and Pietroszek, 2006). Since the concepts of MAIA were introduced in this Chapter, we will also evaluate its completeness and soundness in Sections 3.6 and 3.7.

### 3.5.3 Evaluating MAIA as a Software Tool

Usability, usefulness and parsimony are more on the user level and related to MAIA as a software tool. To evaluate MAIA as a software tool, we use the body of literature for software evaluation where we have identified four different methods:

- 1. Survey: users are asked to provide information about the method or tool (Kitchenham et al., 1997).
- 2. Formal Experiment: users are asked to perform the same task using different methods or tools (Kitchenham et al., 1997).
- 3. Case study: the impacts of the method or tool on real world case studies are studied (Kitchenham et al., 1997).
- 4. Comparison with similar software: the method or tool is compared with other similar approaches using a set of features (Moody, 2005).

To evaluate the usability and usefulness of MAIA we will use all of these methods. We will identify a set of factors that reflect these two features. In Chapter 4, we will also use four real world projects as case studies for MAIA. These case studies will also be used to make MAIA parsimonious by removing the redundant concepts during the development of the simulations.

Furthermore, we will address the second method of evaluation by *redeveloping* ABMs for two of the case studies with MAIA. In Chapter 4, we will also apply the first method of software evaluation by analysing the feedback from the users of MAIA.

IAD concept/relation	MAIA concept/relation		
Physical World	Physical Structure		
Attributes of the Community	Attributes of the agent, ADICO		
Rule-in-use	ADICO institutions		
Action Situation	Action Situation		
Participants	Agents		
Action Arena	Action Arena		
Patterns of interaction	Observed in simulation		
Evaluative Criteria	Evaluative Structure		
Outcomes	Observed from the simulation		
Feedback Evaluative Criteria: Rules,	Make changes in the simulation		
Community, Physical World & Action			
Arena			

Table 3.2 – Concepts that map between the IAD and MAIA framework.

Finally, to evaluate the whole package of MAIA as a simulation platform, we will use the fourth method of evaluation in Chapter 7. For this method, we will define a set of features to compare MAIA with similar ABMS tools.

## **3.6** Compliance with the IAD Framework

To compare MAIA with IAD, we talked to several IAD experts and studied cases that were conducted using IAD. In this section we will compare these two frameworks to verify that the shared components between these frameworks represent similar concepts and that the added details (e.g., physical component) are in line with the ideas in IAD.

## 3.6.1 Comparison between IAD and MAIA: General Concepts

A comparison between the concepts and relations in the IAD framework (Figure 2.1) and the MAIA framework is shown in Table 3.2.

One of the main differences between MAIA and IAD is the concept of community. The definition of community by Ostrom is quite broad. It includes social attributes such as norms of behaviour and demographic properties of participants. The community concept in IAD is spread among two concepts in MAIA. First, those attributes that are more associated to individuals such as demographics are given to the agent concept in MAIA. Second, since we are using the ADICO grammar of institutions that also involves social attributes (e.g., norms, shared strategies and culture) (Ostrom, 2005), we consider this aspect of the community in IAD as part of ADICO in the constitutional structure of MAIA.

One other difference between IAD and MAIA is 'rules-in-use'. Rules-in-use have been classified into 7 types: boundary, position, choice, payoff, information, scope and aggregation (Ostrom, 2005). In addition, Crawford and Ostrom (1995), propose the ADICO structure for the rules in use. Since the purpose of a MAIA model is to produce a simulation, the concepts need to be structured and formal so that they can have a computational meaning. Therefore, we do not use the different types of rules as discussed by Ostrom (2005) and only stick to the ADICO structure that can be coded into a simulation.

Another difference between the two frameworks is 'patterns of interaction'. For studying systems using the IAD framework, patterns of interaction are an animation for the analysis. They are either created by applying game theoretical analysis or using ABMS (Smajgl et al., 2009). In MAIA, patterns of interaction are observed as an outcome of the agent-based simulation. Therefore, we do not consider a representation of this concept in the MAIA framework because it is not modelled in the agent system but is rather an emergent outcome.

The last difference between MAIA and IAD is the type of feedback that goes into other concepts in the frameworks from the evaluative criteria/structure. In the IAD framework, the feedback is used in the actual system under study, while in MAIA, the feedback is used to evaluate the simulation and debug it. Nonetheless, in both cases, the final feedback is meant to provide recommendations for the real system.

#### 3.6.2 Comparison between MAIA and IAD: Action Situations

Since action situations are important and detailed concepts in both frameworks, we compare the internal structure of an action situation in IAD (Figure 3.14) and in MAIA (Figure 3.15). There are more relations between the components of an action situation in MAIA than what is shown in Figure 3.15. For example, an entity action may also be assigned to an agent or a physical component and there are many more factors that link actions to outcomes. However, for the sake of comparison with IAD and to show how MAIA covers all concepts in this framework, we only stick to the aspects in Figure 3.15.

As illustrated in Figures 3.14 and 3.15 the action situations in both frameworks are only different in terminology but the concepts and relations are almost identical. The only conceptual difference between the action situations in IAD and MAIA, is the usage of the model of an action situation rather than the action situation concept itself.

In IAD, the model can be used empirically or theoretically (Ostrom, 2005). In empirical analysis, the actors are studied given their internal and implicit behaviours within an action situation model. The observed interactions and outcomes are analysed according to these actors and the evaluative criteria. Similarly, in theoretical analysis using IAD, predictions about the interactions and outcomes are presented based on the theories used by the analyst to study the model and also the evaluative criteria. A MAIA model is however, treated some what differently. The modeller uses empirical and theoretical analysis to conceptualize the model. The predicted interactions and outcomes are the results of the simulation given the evaluative criteria in the model.



Figure 3.14 – The internal structure of action situations in the IAD framework (Ostrom, 2005).



Figure 3.15 – The internal structure of action situations in MAIA.

#### 3.6.3 IAD Case Studies

We studied research on several cases that were conducted using the IAD framework to find out whether the concepts described in these studies can be captured with MAIA. The cases we studied were: (1) Australia Outback (empirical analysis and ABMs) (Smajgl et al., 2009), (2) Decentralized Forest Governance (empirical analysis) (Andersson, 2006), (3) Fishery Management (Imperial and Yandle, 2005), (4) Forestry Management (a combination of theoretical and empirical analysis) (Coleman and Steed, 2009), (5) Watershed Management (empirical analysis) (Aylward and González, 1998), and (6) Protected Area Management, (empirical analysis) (Togridou et al., 2006).

The information available about each case in the reports (mostly articles) could easily be captured with MAIA as the descriptions were relatively abstract. We particularly tested whether the added details (to IAD) in MAIA can sufficiently capture the case descriptions. The major difference between those case studies and the models we capture with MAIA are the modelling of institutions. Since the IAD case studies mostly focused on theoretical analysis and empirical research, the researchers analysed the institutions in terms of the 7 different types of institutions. However, because our goal is to develop simulations, we use the ADICO structure to formalize the institutions in order to incorporate them in a simulation and do not consider the 7 types of institution. Nonetheless, it would be useful to study the different types of institutions with the simulation outcomes.

One other difference we noticed in the IAD case studies was that the analysis mostly started by identifying the action situations. However, MAIA starts by identifying the agents and only gets to action situations when the institutions and physical components have also been defined. Nonetheless, there is not strict rule for starting from any particular concept in MAIA.

The comparison between the IAD and MAIA frameworks confirmed that MAIA is conceptually as complete as IAD.

## 3.7 Soundness of MAIA

The soundness rules are checked by verifying the MAIA class diagram in StarUML<sup>9</sup>. A well-formed class diagram has a set of classes, a set of abstract classes, a set of associations and a set of superclass relationships between classes. Classes are linked to each other with associations (e.g., agent - physicalComponent). An association has two *associationEnds* which describe the role each class has in that association and the *multiplicity*, describing the number of instances.

There are 38 well-formed rules in StartUML. Below are some examples for the rules that the MAIA class diagram has been checked with:

- 1. All associations have unique names.
- Each AssociationEnd of an Association must have a unique rolename (e.g., agent own physicalComponent).
- 3. No class can be supertype of its subclasses. In other words, circular inheritance is not allowed.

<sup>&</sup>lt;sup>9</sup>http://staruml.sourceforge.net/en/

4. A transitive, reflexive relationship exists between all classes in a diagram.

By verifying the UML class diagram of MAIA, we made sure that there are no inconsistencies in the relations between MAIA concepts.

## 3.8 Conclusion

In this chapter, we presented MAIA as a conceptual framework for ABMS that builds on social theories and frameworks. We defined a set of criteria for evaluating MAIA. Using two of these criteria, we validated the completeness of MAIA and verified its soundness.

By using the IAD as a basis of this framework, the MAIA meta-model facilitates conceptualization of socio-technical systems with the assumption that 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 more realistic agents who 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 to the Simulation Practice** The MAIA meta-model contributes to simulation especially ABMS practice in the following aspects:

- It visualizes conceptualization of socio-technical systems models 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 knowledge of computer science, programming, or even ABMS.
- MAIA provides a precise ontology for IAD concepts to facilitate ABMS of various socio-technical systems.
- The MAIA meta-model language is independent both of programming language (although it might be geared more towards object oriented languages) and application domain, so that the concepts can be documented and reused as required (as pointed out in (Heath et al. 2009)).
- MAIA assists participatory model development<sup>10</sup>. 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.

Besides these intended contributions we see an additional benefit. During system conceptualization, the system boundaries become more clear because MAIA encourages the modeller to think about the concepts that may be relevant to be included in

 $<sup>^{10}</sup>$ MAIA can be used in 'joint design workshops' and 'user panels' which are identified as two of the standard participatory techniques in (Ramanath and Gilbert, 2004).
the agent-based model. With 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 which we will explain in the next chapter, 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.

The MAIA meta-model presented in this chapter explains a social system in a high level modelling language. The model description can be used by a programmer to develop a computational simulation since the structure of MAIA and its concepts are sufficiently informative. However, with the content of this chapter, the programmer needs to rely on her creativity to build a simulation since there are no strict rules for simulation development and there is not software that would automatically translate a MAIA model into a simulation. While this abstractness gives high flexibility to the programmer, in the later chapters we will further develop this research to provide more guidelines for programmers and show how MAIA models can be (semi)automatically translated to executable simulations.

In the next chapter, we will explain the case studies that we have conducted during the development of MAIA. These will be used to evaluate the usability, usefulness and parsimony of MAIA. Theory is splendid but until put into practice, it is value-less.

James Cash Penney

# Case Study Evaluation

# 4.1 Introduction

Software tools and methods are frequently evaluated along two features: usefulness and usability (Landauer, 1996). Kitchenham et al. (1997) identify three different ways of evaluating the usefulness and usability of software methods and tools. First, survey evaluation, where users are asked to provide information about the tool. Second, formal experiments to ask users to perform the same task using different methods or tools. Finally, the impacts of the tools on real world case studies are studied.

Continuing the evaluation of MAIA from Chapter 3, we use four real world projects as case studies for MAIA to evaluate its usability and usefulness. We address the second method of evaluation by *redeveloping* ABMs for two of the case studies (i.e., consumer lighting, wood-fuel market) with MAIA. These two case studies have already been modelled using other ABMS tools. Finally, we analyse the feedback from the users of MAIA (i.e., developers of the case studies) through a questionnaire that we have developed using a set of features.

All the case studies are about the study of the effect of a strategic policy on the behaviour of individuals within a socio-technical system that has institutional influence. These case studies were modelled during the development process of MAIA to actually help build this conceptual framework and its accompanying tool. We removed redundant variables and simplified MAIA during this process to make it parsimonious. For this chapter, we have updated each of these four cases to reflect the final version of MAIA.

In Sections 2-5, we will give a brief overview of each case study and explain why we have selected each one of them. We then briefly describe the conceptual model of each case according to MAIA structures, to show how MAIA concepts are represented in each study. MAIA structures are described under separate heading for each case. By explaining the conceptual models, our goal is to also show how some practical and common aspects of socio-technical systems such as negotiation and cooperation are captured in a MAIA model. The common aspects which we introduce with each case study are defined in grey boxes to explain the general idea without case specific details. We show some simulation results and conclude each case study by discussing lessons learnt and reflecting on the usefulness of MAIA for each of the case studies. Finally, in Section 4.6 we will evaluate the usefulness and usability of MAIA based on the survey that included MAIA users.

# Usefulness and Usability Indicators for MAIA

The literature offers a set of indicators for evaluating the usefulness and usability of software. Considering the purpose of MAIA as a simulation tool, we define a set of usefulness and usability factors (cf. Section 3.5.1):

- 1. Effectiveness: Whether MAIA improves the process of ABMS.
- 2. Cooperation: Whether MAIA facilitates participatory ABMS.
- 3. Applicability: Whether MAIA is applicable to those socio-technical systems where the purpose is to study the effect of long term policies and decisions on the behaviour of individuals.
- 4. Efficiency: Whether MAIA increases the efficiency in developing ABMs.

The usability indicators are defined according to the following:

- 1. Ease-of-use: How easy it is to use MAIA.
- 2. Learnability: How easy it is to learn MAIA.
- 3. Ease-of-becoming-skilful: How easy it is to become skilful.
- Clarity/Understandability: How clear it is to understand the concepts in MAIA.

# 4.2 Case Study I: Consumer Lighting Transitions

The consumer lighting case study is a clear example of a socio-technical system where technological advancements lead to institutional development. The model aims to clarify the effect of long term strategic policies on the behaviour of individuals.

With this case study we have three objectives. First, to find out how straightforward it is to develop a conceptual model of a system with the mentioned characteristics using MAIA. Second, to find out whether the concepts are sufficient to explain a relatively simple socio-technical system. Third, since this case study has been previously modelled (*see* (Chappin, 2011; Afman et al., 2010)), we also want to find out whether the concepts considered in the original model can be captured with MAIA.

#### 4.2.1 Case Overview

Lighting is essential for modern living. Edison's first carbon filament glow bulb had a lifetime of 45 hours and an efficiency of  $2 \text{ lm/W}^1$ . Many gradual improvements in electric lighting technologies increased the lifetime of the bulbs and the electric efficiency (Gendre, 2003). By 1912, the glow bulb's efficiency had improved to reach a light output of 12 lm/W of electricity. Technological progress in incandescent bulbs was halted at that point. Presently, the incandescent lamps are hardly more efficient. Even now, over 98% of the electricity used is converted into heat and not into light.

More energy efficient alternatives have been developed such as the compact fluorescent lamp (CFL) (Azevedo et al., 2009). The CFL was first introduced by Philips in 1980, and offered four times energy savings and a much longer lifetime, with some disadvantages (size, weight). Subsequently, the CFL was much improved in the decades afterwards, and was known as the 'saving lamp'. CFL's offer clear benefits for many applications, and many governments try to stimulate their use (e.g. (Mills, 1993; Martinot and Borg, 1998)), but these stimulus policies have seen limited success. CFL saving bulbs are currently present in only 55% of European households (Bertoldi and Atanasiu, 2007). Another lighting development is solid-state lighting: the Light-Emitting Diode (LED) which continues to achieve significant gains in electric efficiency (Curtis, 2005; Holonyak, 2005; Azevedo et al., 2009).

Consumers have only partially adopted CFL and LED technology because of a number of obstacles (Menanteau and Lefebvre, 2000). First, CFL and modern LED saving lamps are characterized by high up-front cost for consumers and poor light quality. In fact, consumers implicitly use high discount rates when purchasing energy efficient durable goods (Hausman, 1979; Kooreman, 1996). Second, halogen lamps are more attractive because they fit in popular designs and do not have the disadvantages that CFLs have (e.g., cost, unfavourable colour, size, no dimming option).

In consumer lighting, changes are forthcoming. The European Union's phase-out of incandescent lighting is a clear strategy that will change the sector. It involves regulations designed to remove the cheapest forms of inefficient household lighting from stores (CEC, 2009). It is uncertain whether the lighting sector will become efficient overnight; consumers may switch to forms of inefficient lighting that are exempt from the phase-out; or consumer behaviour will change. The precise dynamics induced by the phase-out are unknown.

There are three possible policy interventions which we can experiment: (1) a ban on bulbs, (2) an incandescent taxation scheme and (3) a subsidy scheme on LED lamps. The first policy entails a complete ban on the standard incandescent light bulb.

To get better insights in these dynamics, Afman et al. (2010) developed an agentbased simulation using Java with elements of RePast. This model encompasses consumers that buy lamps, based on the available luminaires in their houses, their

<sup>&</sup>lt;sup>1</sup>Light output is measured in lumen (lm). An ordinary incandescent 75 W bulb (which is now banned in the EU) emits more or less 900 lumen at 12 lm/W. With a theoretical maximum of 683 lm/W, the bulb is <2% efficient (Azevedo et al., 2009).

personal preferences and the preferences of their acquaintances. By introducing policies that affect consumers' behaviour (banning light bulbs, taxing light bulbs, or subsidising energy efficient alternatives), the system outcome is explored over a simulated period of 40 years.

# 4.2.2 Modelling Consumer Lighting with MAIA

In this section we revisit the consumer lighting model using the MAIA meta-model<sup>2</sup>. The description of the conceptual model is made in an effort to be compact, yet highlighting the major concepts in the model. In this case study, we will go into the decision making process of agents in detail to show one way of implementing decisions in MAIA. We also illustrate some of the MAIA tables and diagrams used for this particular case.

#### **Collective Structure**

The collective structure in the consumer lighting sector defines the agents in the model. For this socio-technical system, we define four types of *agents*: the household member, the retailer, the manufacturer and the government. The household member has *properties* including the number of lamps he owns, usage of lamps, total number of luminaires, socket types and neighbours (agents in the same neighbourhood have the same id). Household members have *personal values* such as light colour, colour rendering and light output<sup>3</sup>. Household members form opinions (i.e., *information* attribute of the agent in MAIA) about individual lamp types, technologies and brands. Household members start with neutral opinions to become negative or positive during the simulation.

The consumer's lamp purchase *decision* is modelled using multi-criteria decision making (MCDM), incorporating *criteria* related to *properties* of the lamps (purchase price, efficiency, and lifetime), preferences for subjective lamp qualities (colour, colour rendering index (CRI), light output), and opinions (perceptions) on the lamp's characteristics (lamp model, brand, and technology type). A final criterion relates to what other consumers do (word of mouth).

A number of important behavioural assumptions underlie the *criterion weight* factors that determine the relative importance of the normalized scores. As the purchase price needs to be the most important criterion (Menanteau and Lefebvre, 2000), it is assigned a high weight factor of 4. Then, lamp efficiency, colour rendering, light colour, the household's opinion of lamp technology type and normative adaptation (word of mouth: imitating neighbours) are assigned a weight factor of 2: important, but not as strong as the purchase price. Last, the lamp's light output, lifetime, and the consumer's opinions on brand and lamp model are even less important, these get a weight factor of 1. Between household agents, weight factors differ by +/- 50% to make them heterogeneous (Chappin, 2011).

<sup>&</sup>lt;sup>2</sup>This section is partly published in (Ligtvoet et al., 2011)

<sup>&</sup>lt;sup>3</sup>Agents instances are heterogeneous in their initial portfolio of lamps (total number of luminaires, socket types, and the specific lamps installed initially) and in their personal values for light colour, colour rendering and light output.

A consumer's opinion (a [-1..1] scale) is changed as a result of own experiences with bought lamps and through information received from neighbours (word of mouth). Parameters and increment values used for autonomous opinion change are:

- if positive experience: + 0.1
- if negative experience: -0.3
- if positive experience, contrary to existing opinion: +0.2

#### **Physical Structure**

There are two *physical components* in the physical structure of the consumer lighting system: luminaires and lamps. The *properties* of luminaires are: socket type (E27, GU10, E14), wattage, location, usage<sup>4</sup> and status (operational, failed). The *properties* of lamps are: lamp technology<sup>5</sup>, the expected lifetime, uncertainty of the lifetime, light output, electricity consumption, colour rendering index, colour temperature, voltage, shape, socket, and purchase price. The *behaviour* of a lamp is 'defect'. Both lamp and luminaires are *private* components meaning that they are owned by different household agents. The *affordances* of a lamp are being 'sold', 'bought', 'used' and 'produced'.

#### **Constitutional Structure**

The consumer, seller, producer and rule enforcer are the four *roles* in the system. The *objective* of the consumer is to have pleasant light in his house. The *objective* of the seller and manufacturer is income and the rule enforcer wants reduction of electricity consumption. The *institutional responsibilities* of the agents in the consumer role are 'follow word of mouth', 'buy lamp' and 'use lamp'. The seller role can 'sell lamps' and the producer role can 'produce lamp'. Since we considered a one to one relationship between the agent types and the roles in this case (manufacturer for producer role, household member for the consumer, retailer for seller), and the agents do not switch roles, it is not obligatory to define *entry conditions* for the roles. Table 4.1 shows the *institutions* in the consumer lighting model.

#### **Operational Structure**

In the *action arena*, the *entity actions* are put in seven *action situations* illustrated in Figure 4.1. Each action situation consists of a number of entity actions. For example, lamp replacement situation which involves the consumer role consists of the following entity actions which take place in sequence: lamp defect, check socket type, update opinion (based on the history of this defect) and replace lamp. In this version of the model, households do not change their luminaires. Therefore, the options to change lamps are limited to socket-compatible lamps. Similar to this action

 $<sup>^4</sup>$  We define a distribution function for the place of lamp in the house based on survey data on the number and usage of lamps in consumers' homes (*see* (Afman et al., 2010)).

<sup>&</sup>lt;sup>5</sup>Empirical data on 70 lamp technologies were collected in a variety of lighting stores.

Name	Attribute	Deontic	alm	Condition	Or else	Type
taxing	consumer	must	pay tax	if $lamp$ to $buy = incan-$	not able	rule
				descent	to buy	
subsidy	consumer	may	receive	if lamp to $buy = LED$	I	rule
			subsidy			
word of	consumer	-	buy lamp	if lamp is that of neigh-	I	shared
mouth				bour		strategy
ban selling	seller	forbidden	sell	if ban policy active	fine	rule
			incandes-		selling	
			cent			
ban produc-	producer	forbidden	produce	if ban policy active	fine	rule
tion			incandes-		selling	
			cent			
fine selling	rule enforcer	may	fine seller	if sell incandescent &	1	rule
				ban policy active		
fine produ-	rule enforcer	may	fine pro-	if sell incandescent $\&$	I	rule
cing			ducer	ban policy active		

Table 4.1 $-$
The
Institution
table
for
consumer
lighting

situation, there is the purchase lamp action situation where the actual choosing and buying of the lamp take place according to different policies. When two agents want to perform the collective action of a purchase, a consumer raises a flag indicating that he wants to buy a lamp. A seller searches for raised buy flags in different agents. The seller will then give a lamp to the consumer and add to his money. The consumer will also take the lamp and give away some of his money when it comes to his turn in the next round. When a consumer communicates his opinions, the opinions of a neighbour are averaged between the old value and the other consumer's opinions.

#### Collective Action in MAIA

Agent communication for performing collective action is done by raising flags. When an agent decides to perform an action that requires some other agents' action, he raises his flag for that action. A second agent observes the flag when it is his turn to perform the action. If the second agent decides to perform the action, the statuses of both agents are updated to reflect an executed action, otherwise the flag remains raised.

As illustrated in Figure 4.1, even in the situations where there is, for example, a ban on light bulbs, the producers still get the chance to *decide* to produce and if so, they receive a fine. Therefore, the agents have flexibility for non-compliance within the *institutional structure*. One situation that is somewhat distinct compared to others, is the technological advancement action situation. It is especially important to consider the implementation of change in prices according to technology, later on in the model. The entity action in this situation is 'change price'. Although the prices of all lamp models differ, the lamp technology type determines the decline in price over time. Newer technologies – LED and CFL – are modelled to decline faster than proven technologies (halogen and incandescent). Due to technological progress, the efficiency of each technology increases over time, where the efficiency of more 'modern' technologies CFL and LED is assumed to increase faster than for the old technologies.

#### **Evaluative Structure**

Three main *evaluative variables* were defined regarding the main questions: system level adoption of the technology types, average electricity consumption level and average money expenditure on lamp purchases. A *scope matrix* relates these *variables* to the *entity actions*. For example, there is a *direct relationship* between system level adoption of technology types and the buy lamp entity actions as this is where the actual buying of different technologies happens. There is an *indirect relationship* with all the other entity actions.

The *reality closeness matrix* is a similar matrix where the indicators show how well the simulation is reflecting reality. The number of lamp defects and the price of lamps are two such variables.

Action Situation				Condition	Institution
	Plan type	Entity actions	Loop Condition		
Consumption					
	Sequence	Use lamp, Update opinion			
Lamp Replacement	Sequence	Lamp Defect, Check Socket Type, Update opinion, Replace lamp			
Technological Advancement	Atomic	Change price of lamps			
Purchase with taxa	tion policy			Taxation experiment = true	
	Sequence	Update neighbour opinion, buy lamp LED, buy lamp incandescent, pay tax, sell lamp			
Purchase with ban	policy			Ban experiment = true	
	Sequence	Update neighbour opinion, buy lamp LED, buy lamp incandescent , sell incandescent, fine seller, sell LED			
Purchase with sub	sidy policy			Subsidy experiment = true	
	Sequence	Update neighbour opinion, buy lamp LED, receive subsidy, buy lamp incandescent, sell incandescentlamp, sell LED			
Production with tax	kation policy	/		Taxation experiment = true	
	Sequence	Produce incandescentlamp, produce LED			
Production with ba	n policy			Ban experiment = true	
	Sequence	Produce incandescent lamp, fine producer, produce LED			
Production with su	bsidy policy			Subsidy experiment = true	
	Sequence	Produce lamp, produce LED			

Figure 4.1 – Operational environment for a consumer lighting model.

# 4.2.3 Lessons Learnt

This case study was conducted to find out whether the concepts and relations in MAIA are sufficient to develop an agent-based model of a socio-technical system. Since the modellers of the two versions of the model were different people we took the opportunity to model the system without any previous assumptions. Then, we studied the previous model (in terms of concepts) to explore the differences and find out whether there were any particular concepts that were present in the previous model that could not be captured with the MAIA concepts<sup>6</sup>.

Our study showed that not only all the previous concepts could be captured with MAIA, there were many more options in the new model that were not captured previously, either (1) to keep the model simple or (2) were quite complex to consider. For example, there were no manufacturers in the first version of the model in order

<sup>&</sup>lt;sup>6</sup>Since the primary focus of this case study were the concepts in MAIA, we conceptualized the consumer lighting system and did not redo the programming of this case.

to keep the model simple and manageable. This however, could have influenced retailers' strategies (e.g., marketing). As another example, allowing the agents to produce/sell banned light bulbs was simply complex. However, the structure of MAIA and the availability of the concepts and relations in the modelling language made it quite easy to consider these issues. In addition, some entities in the model such as the government were considered as the environment in the previous model while with MAIA, such elements are considered as agents. This is more logical as it gives flexibility to the definition of the government as an active component.

An important benefit of MAIA was that instead of hard coding policy measures in the program, linking the (formally defined) institutions or policies to governmental enactment facilitated the implementation of different policies in the model and the analysis of their influence on other entities. Furthermore, we were able to model different policy experiments within one simulation by defining different action situations in the action arena as shown in Figure 4.1.

Finally, reflecting on the original consumer lighting model, we cannot prove that using MAIA has improved the model as this is also dependent on the modeller. What we do find is that MAIA can help in the conceptualisation phase and function as a guideline for including necessary elements of socio-technical systems. The MAIA meta-model also contains elements that are not necessary for the purpose of the model. For instance, the links between physical structures (physical connections of houses) are suggested by the MAIA meta-model, but were not modelled. Despite the fact that such links are unnecessary, the meta-model explicates the possibility to model them.

This case study was conceptualized using MS Excel. The concepts in MAIA were put in corresponding tables and diagrams (e.g., agent table) which were filled in by the modeller. However, keeping track of the relation between these tables and diagrams proved to be difficult. Therefore, we saw the need for a tool that takes care of these relations so that the modeller would not have to keep track of every relationship herself.

One other issue that can be considered as a drawback for MAIA, while performing this case study, is that there is no specific architecture for making agents. All MAIA gives to the modeller is the set of attributes that could be considered when defining an agent. These include the properties of an agent, his personal values and his decision making process, among others. However, MAIA considers this information in a set of tables without structuring them into an architecture such as BDI, reactive or some psychological framework. Nonetheless, this can also be considered as a positive point because it gives more flexibility to the modeller in choosing the most appropriate framework.

**Redundant Concepts** The major change we made to the MAIA meta-model after this case study were the conceptualization of agent interaction and the structure of institutions. In the first version of the MAIA, interaction was explicitly defined between agents to follow similar structure in the IAD framework (i.e., patterns of interaction). However, in this case study we realized that agent interaction is implicitly considered in the structure we have defined for actions. Therefore, this definition proved to be redundant and was taken out of the meta-model. Furthermore, in the first version of MAIA, we did not use the ADICO structure to define institutions. Therefore, the description of an institution, in this case the EU policies, were defined in an unstructured text which was not easy to use in the simulation.

**Usability and Usefulness** The MAIA meta-model was effective for building the consumer lighting model because it structured all the required information in a manageable manner which was less complex to use in an agent-based simulation. We could not test MAIA as a participatory tool because we only communicated with the previous modellers and did not have separation of tasks in the modelling process. However, MAIA was applicable to this case study and since the information was captured in tables and diagrams the modelling process was also efficient. Since the modeller of this case study was also the developer of MAIA, we cannot reflect on its usability at this stage.

# 4.3 Case Study II: The Wood-Fuel Market

This case study is another example of a socio-technical system, where the purpose is to study the dynamics of individual behaviour and decision making on the long term social and technical outcomes of the system. The reasons for selecting this case study was that compared to our previous case, (1) the spatial location of the agents was an important element, (2) there were fewer agents and, more complex communication between them, and (3) agents were located in a market situation which is a common environment in many socio-technical systems and thus worth experimenting.

Similar to our previous case study, this case was also conducted with and without MAIA. Our goal again was to find out whether the previously modelled concepts can be captured with MAIA.

# 4.3.1 Case Overview

The amount of wood that can be harvested in a given time theoretically depends on the natural tree growth, technological restrictions, and long-term ecological and economic concerns. However, political concerns and legal obligations result in less harvestable wood quantity which is referred to as annual allowable cut (AAC) (Steubing et al., 2010).

Foresters, forest owners or commercial forestry companies sell energy wood (or wood-fuel) which covers 10% - 50% of the total harvested wood mass consisting of branches and leaves that are not as useful as round-wood to a variety of consumers including households and commercial plants. The amount of wood-fuel offered on the market is therefore, partially a function of the round-wood production. In many countries, private forest owners increase the complexity of the market by resisting to offer their wood in the market due to personal goals such as "aesthetics, recreation and wildlife habitat" which are not fully related to profit maximization. These non-financial motivational factors significantly determine the market actors' relations

(Kostadinov and Steubing, 2011).

The goal of this study is to explore the availability of wood (and wood-fuel) in the Swiss market to find out how much wood is offered by wood producers and assess to what extent this quantity of supply covers consumer demand. Questions to be addressed are: "Which agents get what amount of energy wood at what price? Do changes in foresters' and forest owners' value systems, for instance, a stronger commercial orientation at the loss of traditional social bounds, lead to an increase or decrease in the available amount of energy wood on the market? Can incentives be found to increase the overall amount of energy wood on the market?" (Kostadinov and Steubing, 2011).

A first version of the wood-fuel market model was implemented by Olschewski et al. (2009). The model was able to reproduce plausible market results but little domain knowledge was captured with the model and the agents were extremely simple entities.

## 4.3.2 Modelling the Wood-fuel Market with MAIA

In this section<sup>7</sup>, we revisit the wood-fuel market which is implemented using the MAIA meta-model.

#### **Collective Structure**

There are three different types of *agents*: supplier, consumer and intermediary. These agents may own forests or companies. We define three different *personal values* both for suppliers and consumers: profit, friendship and geographical distance. Consumers make *decisions* about installing a wood energy heating system (market entry) or replacing an existing one with alternative energy sources (market exit). The decision of the supplier agents on wood-fuel production depends on their *properties*, including level of education, age and professional occupation. The decision making algorithm used in this model is not an MCDM but rather a regression-based model called analytical hierarchy process (AHP)<sup>8</sup>. Similar to MCDM, AHP associates weights to rank possible alternatives (Saaty, 2008). The agents make decisions about producing, selling and buying, using the weights given to each agent based on his personal value. Table 4.2, gives an overview of this weighting scheme. Agents are either standard, profit-oriented, friendship-oriented or distance-oriented depending on the values they have for each of the three personal values. Based on this classification, different weighting schemes are given to the agents.

#### **Constitutional Structure**

The identified *roles* in this case study are foresters, private forest owners, small private energy wood consumers, public wood-fuel consumers (municipalities), commercial energy wood consumers, district heating network operators, forestry companies, bundling organizations, sawmills and finally small private heat consumers.

<sup>&</sup>lt;sup>7</sup>This section is partly published in (Kostadinov et al., 2012; Steubing et al., 2011).

<sup>&</sup>lt;sup>8</sup>MAIA has a place-holder for agent-decision making. The modeller has the freedom to use (i.e., plugin) any decision algorithm that best suits her purpose.

Types	Weight pro	ofit Weight	friendship	Weight distance
	criterion	criterion		criterion
Standard	0.6	0.3		0.1
Profit oriented	0.9	0.05		0.05
Friendship oriented	0.05	0.9		0.05
Distance oriented	0.05	0.05		0.9

Table 4.2 – The weights used in the decision making of agents based on their personal values (Kostadinov et al., 2012).

More details about the roles and institutions can be found in (Kostadinov et al., 2012).

# **Physical Structure**

The *physical components* are tree, energy wood, round-wood. Since the geographical location of agents is an important element in this model, we also define forests and companies as physical components. Amongst other *properties* which are similar to our previous case study, each forest or company has a X-axis and a Y-axis *property* to show the location of these two entities in the model. To capture the distance between various buyers and sellers which in fact affects their business relationship, we model the *connection* between companies and forests. The network of buyers and sellers based on these connections is later illustrated as a simulation result in Figure 4.3.

# **Operational Structure**

We define four *action situations* in the wood-fuel arena: production, round-wood market, wood-fuel market and consumption.

- 1. Production: Agents decide to *produce round-wood* by looking at the demand (it is a demand-driven process). They also look at the AAC rules to *calculate* how much wood they can cut. Wood-fuel production depends on the amount of round-wood produced. It is also partially dependent on the maintenance activities of foresters and the manufacturing process of the sawmills which produce timber.
- 2. Round-wood Market: In this action situation agents sell and purchase roundwood. The sawmill agents and wood trader agents sell round-wood to private forest owner, forester and wood trader agents.
- 3. Wood-fuel market: In this action situation, agents sell and purchase wood-fuel. Small private wood-fuel companies, public wood-fuel consumers, commercial consumers, heating operators and pulpwood consumers sell wood-fuel to wood traders, private forest owners, foresters and sawmills. The price of wood-fuel is influenced by a number of external variables: oil & gas, electricity, pulp & paper and the timber product market.

4. Consumption: In the consumption action situation, the buyers of wood-fuel (mentioned in the previous action situation), consume the wood-fuel they have purchased depending on their size and consumption.

The number of roles that the agents could take is relatively high. In general, each agent type can *enact* the following roles according to the *entity action* he is involved in:

- Suppliers of energy wood: foresters and private forest owners.
- Consumers of energy wood: small private energy wood consumers, public wood-fuel consumers (municipalities), commercial energy wood consumers, district heating network operators, pulpwood consumers and small private heat consumers.
- Intermediates: forestry companies, bundling organizations, sawmills.

#### Negotiation as a form of collective decision making in MAIA

There are various collective decision making templates that can be plugged into MAIA models as action situations. One of the most common forms is the negotiation process present in every market situation. Similar to collective actions, the negotiation procedure is also based on flags. However, both parties must agree on a price before an action (e.g., sign contract) is executed. In its simplest form, there is only one round of price exchange. However, the negotiation can also take place over many offer rounds.

To show how a collective decision takes place in a MAIA model, we elaborate on the negotiation process in the market situations:

- 1. Every consumer agent (i.e., buyer) has a list of all the seller agents. When a buyer decides to buy wood-fuel, he takes the first seller in his list and saves the id of that seller as the requestId. This can be viewed by all the other agents in the simulation.
- 2. The buyer calculates the price he is willing to pay and sets his requestPrice which is also observable by others.
- 3. Each seller searches the requestIds of other agents and makes a list of those agents who have his Id as the requestId. He orders the requests starting from the highest offer.
- 4. One by one, the seller checks if he has enough product for the request and if the price is higher than the price offered by the buyer. He accepts or declines the request.
- 5. The buyer checks to see if his offer has been accepted. If so, he signs a contract with the first offer on his list and deletes other requests. If all his offers have been declined, he recalculates his offer and the process starts again.

#### 4. Case Study Evaluation

Figure 4.2, illustrates the *entity actions* and *plans* for the negotiation process in the market situation. This is a fairly simple negotiation process which can be replaced with negotiation libraries that have a higher number of offer iterations (e.g., GENIUS negotiation platform (Hindriks et al., 2009)).



Figure 4.2 – The market situation illustrating the negotiation process. The dotted circles show plans and the inner grey circles show entity actions.

#### 4.3.3 Simulation Implementation

The programmer of the wood-fuel market model was not the MAIA developer. Since the implementation rules were not extracted yet, the programmer used the conceptual MAIA model to develop the simulation using his own experience. In addition, he used UML interaction and sequence diagrams to add another layer of detail to the MAIA model before starting the implementation.

The simulation was split up into several sub-processes, one for each action situation. Each loop represented one year's execution time, where the agents entered and exited the market. However, there was another inner loop that repeated the roundwood market, the wood-fuel market and the evaluation process 12 times (i.e., month) per year. Furthermore, yet another inner loop repeated each market 6 times in each month (Kostadinov et al., 2012).

## 4.3.4 Simulation Results

By varying wood demand and wood supply, some experiments were conducted. These experiments confirmed some basic economic assumptions on markets: scarce supply leads to higher prices and lower consumption and in an excess supply situation prices tend to fall and new consumers tend to enter the market.

Going into more detail, in extreme scarcity, it can be deduced that pulpwood consumers are more vulnerable than other classes of wood-fuel consumers. It is also concluded that different stakeholder groups can have competing interests and often



Figure 4.3 – The trading relation between different agent types. The circles, triangles and squares represent different agents types. Arrows indicate the direction of the trade from the seller to the buyer (Kostadinov et al., 2012).

not a single optimum situation exists for a given problem. Wood-fuel consumers might be interested in a market structure with a large number of sawmills because this increases wood-fuel availability. However, higher number of sawmills also increases competition amongst round-wood consumers which might be contrary to their interests. Figure 4.3 shows a screen shot of trading relations in a simulation run at a certain point in time.

# 4.3.5 Lessons Learnt

The goal of the wood-fuel case study was to identify the added value of using MAIA by redeveloping a previously implemented model. This case study was particularly interesting to us for the complexity of agent behaviour and decision making. In this case, we had the possibility to explore complex role enactments as there were numerous roles that each agent had the potential to take. The previous model of this case study simply relied on standard microeconomic assumptions. With the new model, the modellers were aiming to find out whether it is feasible to model complicated agent interaction such as negotiation. Furthermore, this model required a spatial representation which was also captured by using MAIA.

Our modelling practice not only confirmed the feasibility of defining agent negotiation and spatial representation with MAIA, but we were also able to verify a especially useful benefit of MAIA. We realized that communication with domain experts becomes relatively easy because the modellers now have an explicit representation of the conceptual model they are picturing in their minds. In this case study, the modellers had several workshops with the expert to verify their conceptual model unlike the previous version of the wood-fuel market model. Furthermore, collaboration was easier between the developers. One of the developers had more knowledge and background in computing while the other was a researcher in the forestry domain. MAIA provided a means for them to discuss and collaborate at a more professional level, with a more elaborate division of tasks.

A limitation of MAIA for this case study was that since the modellers were developing relatively complex agents, they required more detailed and specific agent architectures such as BDI (Rao and Georgeff, 1995). On the other hand, since MAIA did not restrict the modellers to any particular agent architecture or psychological theory, they had the flexibility to use the architecture or theories that best suited their purposes (e.g., (Saaty, 2008)) while using the information they had captured with MAIA. The abstractness of the meta-model also provided the flexibility for them to use their own diagrams and figures where required. For example, they used UML interaction and sequence diagrams to elaborate on interaction and timing, while still following the MAIA structure.

**Redundant Concepts** In the version of MAIA that was used to develop this model, agents had costs and benefits and characteristics as attributes. The cost and benefit concept was not used in the simulation because it was not defined in relation to an entity action. Therefore, we decided to consider the costs and benefits in the decision making process of the agent which is in fact an association between agents and actions. The characteristic attribute (e.g., happy) was a redundant concept because it was implemented in the same manner as any other property (e.g., age), hence, we omitted this attribute. We also renamed *preferences* as *personal values* because preferences are more dependent on the action while personal values are related to the characteristics of agents. Consequently, they can be more generically considered in every decision making process.

**Usefulness and Usability** The MAIA meta-model was both effective and applicable in this case study according to its users. MAIA as a participatory tool was put to test since there were three different people involved in the modelling process and the model was shown to domain experts in several workshops.

The users of MAIA found that it is rather difficult to learn the concepts in MAIA but once they are through that process, becoming skilful is fairly easy. Nonetheless, the concepts were clear and understandable. Since there was no tool support for the development of this case study, using MAIA in excel sheets was quite complicated as many consistency checks had to be done by hand.

# 4.4 Case Study III: E-waste Recycling

The computer waste recycling sector is another system where the social and technical aspects are intertwined resulting in a complex system (Ghorbani et al., 2011). This recycling system in India is a system with many policy problems and limited available data. With this study we aim to find out how much MAIA can help to define a model

and its scope where there is scarcity of data. Furthermore, formal and informal institutions play a great role in this system and are therefore, essential to capture in the model. To explore participatory model development in this project, the domain expert (Sathyam Sheoratan a master student in Industrial Ecology<sup>9</sup>) models the system and verifies the model with the problem owner which is a recycling company in Belgium (Umicore). The programmer is yet a different person (Amineh Ghorbani) who will be building the simulation.

#### 4.4.1 Case Overview

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). Recyclers dismantle computers, refurbish the refurbishable parts and extract precious metals from the circuit boards. These activities take place in a social context with many unwritten rules, norms, and shared strategies among the agents, which are not easy to understand or capture. For example, people with higher skills form recycling units and hire workers based on their family relationships and ethnicity amongst other factors.

A patronage system is often present in recycling units which implies that the employees feel indebted to their boss in one way or another (Huysman, 1994). The result is a complex situation where workers do not have the freedom to leave a unit, and the boss has power over his workers, keeping them poor.

Child labour and unsafe extraction (causing health and environmental hazards) are major growing problems in this sector. One reason for recyclers to stay informal and not register their business with the government is to avoid the risk of inspections by the government which would prevent them from illegal practices. However, there is also a high level of corruption in the law enforcement bodies (Schluep and Programme, 2009).

The Indian government wants to introduce professional recycling companies that take over the precious metal extraction. They would take the dismantled parts from the backyard recyclers, extract gold, and return money equivalent to the value of gold to the backyard recyclers. This would prevent unsafe extraction.

Data shows that the income received from professional companies is higher than the value of the inefficiently extracted gold which would increase the welfare of the recyclers, diminishing child labour. However, the revenues obtained by exploiting children and using unsafe chemicals is higher than the income from the professional companies. Therefore, this policy has not yet been successful.

One of the main questions that we aim to answer with this model is: How would rules for fining recyclers for child labour and harmful extraction influence the economic situation of the recycling units and increase their incentive to work with professional refiners.

<sup>&</sup>lt;sup>9</sup>This case study is conducted as a M.Sc. thesis project (Sheoratan, 2011).

# 4.4.2 Modelling the E-waste Recycling System with MAIA

This system has not been previously modelled and there is very little information about the situation. Therefore, the challenging factor of this case study is to collect information about the system. We use MAIA concepts to select information from the real world system to put into the agent-based model. To show a more detailed overview of our information collection process, we describe the concepts in more detail for this case study.

# **Collective Structure**

In the model, four different types of *agents* are defined: worker, government official, refining company and dealer. We also define a recycling unit which would represent a group of workers. The description of each of these agents is given below:

• Worker: In the model, the workers can take different roles depending on their skill and employment status. The workers belong to different families. To earn an income, they start dealing in e-waste products, and when they can hire other workers from their own family, a network of recycling units appears. The *properties* of the worker agent include: age (child/adult), money, productivity (kg/tick), skill, familyId, groupId, isUnitBoss (true/false), isRegistered (true/false), isEmployed (true/false). They have risk and wealth as their *personal values*.

The workers make *decisions* about:

- whether to fire an employee (if they are unit boss)
- whether to produce e-scrap
- whether to register with the government
- whether to employ children. Both formal and informal unit bosses can decide to employ children. The government can fine them for unsafe extraction if they are formal.
- whether to do safe extraction. Both formal and informal unit bosses can decide to do safe extraction. The government can fine them if they are formal.
- Government official: The government official has a list of registered recycling units. He randomly inspects the formal recycling units. If there is child employment, or unsafe extraction, the government official fines the unit boss.
- Refining company: The Refining company is a big foreign company that can efficiently extract gold and other valuable metals from e-scrap. The *properties* of the refining company are: capacity (kg/tick), money, payOutTime (ticks).
- Dealer: The dealer is the provider of old computers, and is the buyer of the end products: gold, refurbished parts and waste. He is in fact a collection of various local markets. This is done to simplify the system and help keep an overview of the flows in the model.

• Recycling unit: A recycling unit is a combination of workers with different expertise. A boss is at the head of the unit. The unit boss takes care of the deals, and the workers do their job and get paid. Each recycling unit has a groupId and a set of physical components (i.e., computers, refurbished parts, waste, escrap, gold and connector/PWB.

#### **Constitutional Structure**

The *roles* in the model are: unit boss, segregator, reburbisher, extractor, professional end refiner (PER), e-waste dealer and the rule enforcer. The unit boss is the head of his recycling unit. Every adult worker in the model starts as a self-employed worker in the field and is therefore his own unit boss. He can buy and sell products to earn a profit. He can also hire other workers to work for him, and fire them if he does not have enough money to pay them. When the unit boss is hired himself, he loses the role of unit boss, and becomes a segregator, refurbisher or extractor depending on his skills. A unit boss hires and fires employees and manages production in the recycling unit. A segregator can segregate old computers. From computers he gets refurbishable parts, connectors/PWBs and waste. If a recycling unit is formal, he can cooperate with the refining company by sending e-scrap in containers to them instead of doing extraction himself. The unit boss will receive income for the e-scrap. A refurbisher changes refurbishable parts into refurbished parts. An extractor extracts gold from the connectors/PWBs. The rule enforcer registers unit bosses who can then cooperate with the PER. The rule enforcer can inspect formal recycling units to see if they follow the *institutional rules*. If a recycling unit does not follow the rules, the rule enforcer may fine the unit boss.

The defined *institutions* for the e-waste recycling scenario are illustrated in Table 4.3. Patronage is embedded in the model, because workers do not have the freedom to quit their job. Family relation is also considered in the model; agents only deal with family members.

The *dependencies* between the different roles are as follows:

- The unit boss is dependent on the E-waste dealer for profit.
- The unit boss is dependent on his employees for profit. The employees are in turn dependent on the unit boss for survival.
- The unit boss is dependent on the PER for profit.
- The PER is dependent on the unit boss for his e-scrap supply.
- The rule enforcer is dependent on the unit boss for hazard reduction.

#### **Physical Structure**

For the model, the products and their material composition are simplified since the aim of the case study is not to model the flow of electronics and their exact composition. There are 7 types of *physical components* in the model. These are: old computers, refurbishable parts, refurbished parts, connectors/PWBs, gold, e-scrap and waste. All these components have weight and price as their *properties*.

Table 4.3 – The Institution table for the e-waste recycling system



Figure 4.4 – The composition diagram showing the composition of resources in the model.

- Old computers: Old computers are the basic resource in the model bought by unit bosses and sold by e-waste dealers. A segregator can segregate old computers into refurbishable parts, connectors/PWBs and waste.
- Refurbishable parts: A refurbishable part can be refurbished by a reburbisher into a refurbished part and waste. Both refurbishable parts and refurbished parts can be bought and sold by a unit boss.
- Refurbished parts: Refurbished parts are sold by unit bosses to e-waste dealers.
- Connectors/PWBs: Connectors/PWBs contain gold and waste. They are treated by extractors.
- Gold: Gold is extracted from connectors/PWBs.
- E-scrap: E-scrap is for this model technically the same as connectors/PWBs but only sold to the PER. Container cost is a property of this physical component.
- Waste: Waste is the material that is not refurbished and not treated. However, it can still be sold to the e-waste dealers.

The *composition* diagram in Figure 4.4 shows that in the model, old computers can be segregated into refurbishable parts, connectors & PWBs and waste, refurbishable parts can be processed into refurbished parts and waste, and connectors & PWBs can be refined for their gold and have waste as by product. E-scrap is mentioned with a dashed line, since it is an option for the extractor to make e-scrap instead of connectors & PWBs.

#### **Operational Structure**

The *action arena* of the e-waste recycling sector contains the following *action situations*:

- Price Fluctuation: Prices and values of products (e.g., gold, refurbished parts and waste) change over time.
- Hiring and firing workers: unit bosses decide whether they need employees. They can hire a segregator, refurbisher or extractor only from their own family. They fire employees if they do not have enough money to pay their salaries.
- Registration: The unit boss decides to stay informal or become formal. All workers in the model start in the informal sector. When registering, a unit boss has to pay an investment cost. While being formal, he also has to pay a tax every tick. A unit boss may choose to unregister.
- Market: A unit boss can buy old computers, refurbishable parts and/or connectors/PWBs from the e-waste dealer. A unit boss can sell refurbished parts, gold and waste to the e-waste dealer.
- Crystal project: The unit boss has to decide whether he will cooperate with the Professional End Refiner. This means that he will make e-scrap instead of connectors/PWBs during the *action situation* 'Treat old computers', and will sell the e-scrap to the Professional End Refiner.
- Treatment: Segregators in a recycling unit take old computers and segregate these. Refurbishable parts are processed by Refurbishers. Extractors extract gold out of connectors/PWBs.
- Safety Inspection: The government performs an inspection on the formal recycling units. Any unit that performs unsafe extraction or employs children is fined.

Table 4.4 depicts some *role enactments* in the model. It shows in which *entity action* an *agent* takes which *role*. For example, to hire workers, the worker agent has to take the role of a unit boss.

# Agent interaction in MAIA

Agents interact as social entities. Instead of message passing, in MAIA, agent interaction is through shared space where agents can act (produce modifications) and perceive (the modifications) (Pavon et al., 2008). This choice is based on two reasons. First, it requires less complex interaction protocols amongst agents, which may be too sophisticated for the purpose of simulation especially for non-experts in programming. Second, agent-based models are built from bottom-up. This form of interaction provides more flexibility for this bottom-up approach because message passing creates a form of predefined network between agents, which is not desired for this type of simulation. Therefore, the agents have the freedom to communicate with other agents based on their own interests, the institutional constraints and their perception of the simulation environment.

Agent	Entity Action	Role
Worker	HireWorkers	Unit boss
Worker	FireWorkers	Unit boss
Worker	Register	Unit boss
Worker	BuyProducts	Unit boss
Worker	CrystalProject	Unit boss
Worker	SellProducts	Unit boss
Worker	DismantleComputer	Segregator
Worker	Refurbish	Refurbisher
Government Official	SafetyInspection	Rule Enforcer
Government Official	Fine unit boss	Rule Enforcer
Worker	ExtractGold	Extractor
Dealer	SellOldComputers	E-waste Dealer
Dealer	BuyProducts	E-waste Dealer
Recycling Company	TakeE-scrap	PER
Recycling Company	ProcessE-scrap	PER
Recycling Company	PayRecyclers	PER

Table 4.4 – Part of the role enactment table for the e-waste recycling system

#### Evaluative structure

The *scope matrix* in Table 4.5 helps analyse the outcomes of the model. To limit the size of the matrix, we illustrate the relation between only a number of *entity actions* and *problem domain variables*.

As an example, the number of refurbishers in a unit is *directly* (indicated with a 'd' in the matrix) influenced by the 'hire workers' entity action, and *indirectly* (indicated with an 'i' in the matrix) influenced by the 'buy products' action. When buying products, a unit boss may buy refurbishable parts, and in the next round when hiring workers, discover that he needs more refurbishers in his recycling unit and hire a refurbisher. All variables listed in the matrix have at least one direct relation and possibly more direct or indirect relations. Each of these variables are *calculated* per tick. For example, the number of refurbishers in each unit is the sum of those agents in this unit who are enacting the role of a refurbisher.

The *validation* matrix is illustrated in Table 4.6. This is used during the debugging of the program. For example, if the values of the average negotiated prices in the model turn negative, there must be something wrong in either the buy products action or the sell products action. The money of agents is influenced in almost every action situation, and the unit members only in the hire workers action. This way, if some of the values show irregular or illogical behaviour, this matrix helps to track down in which action situation the error could be found.

	Fluctuate Prices	HireWorkers	Register with government	Buy products from Market	Buy products from units	Sell e-scrap to PER	Treat old computers	Treat refurbishable parts	Treat connectors and PWBs	Sell products	Fine unit bosses
no. of refurbishers in unit		d		i	i						
no. of segregators in unit		d			;						
money (unit boss)	i	u	А	d	d		d	d	d	d	d
money (worker)	1		<sup>u</sup>	<b>–</b>	L.		d	d	d	<sup>u</sup>	ů
no. of recycling units		d									
no. of registered			d								
gold/waste/refurbished in World market										d	
e-scrap for refiner						i				d	
e-scrap per unit						i	d				
no. of crystal contract						d					
		1									

Table 4.5 – Scope matrix for the e-waste recycling model.

Table 4.6 – Valiation matrix for the e-waste recycling model.

	Fluctuate Prices	HireWorkers	Register with government	Buy products from Market	Buy products from units	Sell e-scrap to PER	Treat old computers	Treat refurbishable parts	Treat connectors and PWBs	Sell products	Fine unit bosses
Money	i		d	d	d		d	d	d	d	d
no. of product per unit					d		d	d	d	d	
no. of resources per unit				d	d		d	d	d	d	
World market values	d				Ι.					d	
Average negotiated prices					a					d	
composition of units		a				a					
Unit members						a a					
Weight of products per unit		u		d	d		d	d	d	d	
weight of products per unit				u	լս		u	u	լս	լս	

## 4.4.3 Simulation Implementation

To initialize the simulation, we used the literature to set the initial values for various parameters such as number of agents, gold price, average computer weight etc. We also added some variables to calculate and answer the research questions. These variable included average negotiated price for refurbished parts, connectors and escrap, the cumulative amount of refurbished parts and the cumulative amount of e-scrap.

The time loop which repeated all action situations in the action arena was set to 100. In each tick, the agents first checked the entree condition to see which role they can take and what actions they are capable of performing in the current tick. Ordered randomly per tick, the agents entered and existed actions one at a time.

While some analysis variables were calculated or stored per tick, others were calculated at the end of the time loop. The data was stored in CSV files which was then loaded in R and analysed using SQL and LHS (Sheoratan, 2011).

#### 4.4.4 Simulation Results

The model was developed as an effort to answer: "how we can understand the factors influencing the transition of the informal recycling sector in Bangalore into a system cooperating with a professional end refiner." Our goal was to find out whether a relationship between local recycling units and the PER is formed.

An experiment of 100 runs was done with 5 repetitions. In this experiment both economic factors and social factors were tested by varying 9 variables: (1) Number of worker agents, (2) Number of families, (3) Initial money that workers have at the start of a run, (4) Tax that formal recycling units have to pay, (5) Investment cost needed at the time of becoming formal, (6) Corruption, (7) E-scrap price, (8) Container cost for shipping e-scrap, and (9) Time it takes before e-scrap producing units are paid by the PER.

Figure 4.5 shows the fraction of units that chose to do crystal in the model per run (the runs are ordered by the crystal result in this graph). In nearly 80% of the runs a number of unit bosses in the model made the decision to do crystal.

The conditions for doing crystal include high e-scrap price, low investment cost and low taxes. However, this does not imply that those who decide to invest actually make e-scrap because they may not have enough capital to actually buy computers, hire workers and produce e-scrap. In Table 4.7, five areas are characterized by the results of t-SNE graphs (for more information see Sheoratan (2011)) for the amount of e-scrap that is actually produced and sent to the PER. With this table, we can describe the areas, and try to find indicators and characteristics for each.

Area 1: High cooperation with PER This area describes an ideal situation where there is a high starting capital, low tax, low investment cost and high e-scrap price. Even though the container cost is high, the workers are still motivated to become formal. These very favourable conditions in reality would suggest that the government needs to be very cooperative to this sector (low tax and low investment cost) and the professional end refiner should offer a



Figure 4.5 – Standard deviation over the different runs in the e-waste recycling case study (Sheoratan, 2011).

Name	Area 1	Area 2	Area 3	Area 4	Area 5
Outputs					
Computers bought	High	Medium	Low	High	Low
Gold sold to Market	Medium	Medium	Low	High	Low
E-scrap sold to PER	High	Medium	Medium/low	Medium	Low
Average unit size	Medium	Medium	Medium	Medium	Medium
Ending of model	Medium	Low	Low	High	Low
SafeExtraction	Medium	Medium	Medium	Medium	Medium
ChildEmployment	Medium	Medium	Medium	Medium	Medium
Formal/informal ratio	High	Medium	Medium	Low	Low
Inputs					
Number of workers	High	High	High	High	Medium
Number of families	High	High	High	High	High
Initial money	High	High	Medium /low	High	Medium/low
Corruption	Average	Average	Average	Average	Average
Tax	Low	High	High	Medium	High
E-scrap price	High	High/medium	High/medium	Medium	High/medium
Investment cost	Low	Low	Medium/low	High/medium	Medium/low
Container cost	High	High	High	High	High/medium
Pay out time	-	-	-	-	-

Table 4.7 – Analysis of 5 areas in the t-SNE graphs (Sheoratan, 2011).

high price. Also, the workers have a high starting capital, so this is likely to be a scenario for successful local recycling units.

- Area 2: Medium cooperation with PER In this area less e-scrap is sold, and less units are formal than in area 1. There is high tax, nonetheless the low investment cost and relatively high e-scrap price cause several units to become formal. We could view this system as one where the investment cost is subsidized by the government or an NGO.
- Area 3: Very low cooperation with PER In this area, conditions are not favourable: high tax and medium investment cost. Still some recycling units become formal. Perhaps this area describes the minimum requirements or the

borderline-case: the medium investment costs are low enough for some units to become formal.

- Area 4: High continuous activity This area is characterized by high activity. Many computers are bought, and a lot of gold and e-scrap are sold. There is a high starting capital that enables this high activity. Since becoming formal is not directly favourable (medium tax, relatively high investment cost, medium e-scrap price), the majority of workers stay informal.
- Area 5: No activity and high poverty In this area, the workers do not buy products and therefore do not produce much gold or e-scrap. They start with a low capital, which may thus reflect a system with poor workers. The high tax does not encourage the workers at all to become formal. The quick termination of the model may be linked to these factors, and also to the fact that fewer number of worker agents are present in the model.

In general, the model showed that in the case of financially unfavourable settings, the agents mostly stay informal. In fact, the informal sector, by avoiding extra costs to become formal, is more stable and able to survive longer. If circumstances are economically favourable to become formal (low tax, low investment cost, high escrap price, high starting capital), agents are likely to become formal but are still not stable.

# 4.4.5 Lessons Learnt

The goal of this case study was to find out whether MAIA can help build an agentbased model of a system where there is scarcity of information. We used MAIA as an information gathering tool and only collected data on the concepts available in MAIA which helped us narrow down the scope of the system.

The modeller worked with a professional recycling company (UMICORE as one of the stakeholders) to gain insight into the problem and continuously verified his understanding of the system by showing the conceptual model to the company. Furthermore, the programmer was able to make a simulation without having any knowledge about the system and by only relying on the MAIA model and asking for clarification from the modeller a few times.

With this case study, we realised that understanding the distinction between agents and roles may not be straight forward for modellers. However, once this is understood, it is a practical and useful concept of the framework. As the modeller pointed out: "The structuring of the system, identifying roles and institutions through MAIA and detailing the actions in the model were at least as useful as the results for understanding - or identifying - important factors of the system. We could say that the modelling process is a tool in itself for understanding possible factors for change in the recycling sector".

**Redundant Concepts** After conducting this case study, only one redundant concept, 'group', was removed. We modelled hierarchy of agents (i.e., recycling

unit agent containing worker agents) instead of using the 'group' concept which was previously defined as a collection of agents.

Similar to previous case studies, the evaluative structure was not used during model development and only used for later analysis. Therefore, for each variable in the evaluative structure, we updated the structure to include a calculation concept that defines how the outcome variable will be calculated. We also added a number of independent variables to use in the calculation and keep track of, in order to study their relationship with the output variable (e.g., in diagrams).

**Usefulness and Usability** MAIA was effective in this case study because there was scarcity of data and the modeller required some blueprint for the information he needed to build an agent-based model. Participatory model development was executed using MAIA, as the modeller and the programmer were different people and their means of communication was MAIA. In addition, the model was used for verification with domain experts. Furthermore, this framework proved to be applicable to this case study and increased the productivity by facilitating the development of a conceptual model of a system that had not been modelled before or even described in detail.

The modeller found MAIA relatively easy to learn and use even though there were some difficulties in understanding some concepts. Nonetheless, the modeller became quite skilful towards the end of the project and believed that he would be able to use MAIA with its full potential if he were to develop more models.

# 4.5 Case Study IV: Manure-based Bio-gas Production

The manure-based energy system case study forms around a situation where a new technology is introduced into a socio-technical system (Ghorbani, Aldewereld, Dignum and Dijkema, 2012). Besides the technical aspects, the formal regulations that accompany this new technology, influence the system at individual and aggregate level. Therefore, the goal is to explore the feasibility of formulating these regulations into the ADICO structure.

This final case study is selected with the purpose of also exploring the possibilities to incorporate quantitative and qualitative data into an agent-based model. The modeller has gathered qualitative data through interviews with stakeholders and there are also detailed statistical data available for populating the model. Similar to the previous case study, the modeller (Femke de Korte a master student in SEPAM<sup>10</sup>) and the programmer (Amineh Ghorbani) are two people with different expertise who collaborate in developing this model using MAIA.

# 4.5.1 Case Overview

Manure is used as a natural fertiliser because it contains many valuable minerals. However, excess manure is a major problem for animal farmers in the Netherlands.

<sup>&</sup>lt;sup>10</sup>This case study is conducted as a M.Sc. thesis project (De Korte, 2012)

Due to intensive livestock farming, the manure production exceeds the local demand for manure-based fertilisers. Currently, farmers deal with the excess manure by distributing the manure on other lands through intermediaries, with high transportation costs. Policies regarding farming activities including fertilising of land and distribution of manure are rapidly changing, makings farmers unsure of their future.

Considering the availability of intensive livestock farming and excess manure, the province of Overijssel, the Netherlands, wants to increase the share of renewable energy through a biogas infrastructure. In a Manure-based Energy System (MES), animal farmers can produce bio-gas through an anaerobic digestion process which can then be fed to gas pipelines (Morgenstern and De Groot, 2010). However, a biogas network is not regulated within the Dutch Gas Act, leaving uncertainties with respect to its governance.

Biogas production is not feasible without subsidy due to high investment costs. This subsidy, however, is regulated by complex legislation, which has been subject to change, generating more uncertainties. Nonetheless, because of the high potential in producing renewable energy, local farmers are confronted with a decision to be involved in energy production. Since the prospects are unclear, farmers are hesitating to participate in local renewable energy production. Furthermore, while MES can be a source of income, farmers in the Netherlands do not have enough incentive to invest in 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 evolving manure distribution system is complex as changes in the institutions result in different behaviours among the farmers. The introduction of manure-based energy production will only further complicate this evolving socio-technical system.

To understand whether manure can be a source of energy production, it is necessary to learn how manure is currently used and valued by the local farmers within Salland. The purpose of this case study is to identify the long term prospects of using MES in order to create incentives for the farmers to invest in this technology.

# 4.5.2 Modelling the Manure-based Energy System with MAIA

The major differences between this study and the previous ones are that this case study has available quantitative and qualitative data, and the number of actors is fairly limited. This model requires more complex agent behaviour and decision making process which we set out to explore with MAIA.

#### **Collective Structure**

We define three different types of *agents*: animal farmer, intermediary and artificial fertiliser supplier. These agents have *properties* such as money, neighbourhood id, and technology (true,false). The animal farmer owns cattle or pigs. He makes *decisions* about the distribution of manure (or digestate), the investment in ME production and the abandonment or the expansion of his farm. Figure 4.6 depicts the investment decision as a MCDM algorithm. Table 4.8 shows the values that are applied to the *weights* and the *decision threshold*. Based on the outcome of

the MCDM, an animal farmer might decide to invest in technology and become a producer. Since bio-gas technology is expensive and only large scale farms can afford the investment, we also explore the possibility of sharing the technology. Therefore, to explore the effect of cooperation, the decision making algorithm is extended with the decision to cooperate as shown in Figure 4.6.

The intermediary collects manure or digestate (a by product of MES) from the region and distributes these products according to the mineral need within the region. He has *information* about which animal farmers have available manure or digestate and which farmers need minerals. He receives money for his activities from animal farmers. The artificial fertiliser supplier is a company that sells artificial fertilisers.



Figure 4.6 – Decision to produce green gas (De Korte, 2012)

#### Constitutional structure

The three main *roles* in the system are landowner, producer and distributor. The landowner owns crop and grassland. Since crop- and grassland should be fertilized each year, the landowner calculates how much nitrogen and phosphate it needs. The *entry condition* for an agent to take the role of a landowner is to own land. Therefore, farmers who have pigs cannot become landowners because they do not

Weight: Gain	[0.2-0.9]	Default: 0.8
Weight: Character	[0.3-0.6]	Default: 0.7
Weight: Capital	[0.01-0.8]	Default: 0.5
Weight: Neighbour	[0.1-0.4]	Default: 0.3
Decision Threshold	[0-7]	Default: 6

Table 4.8 – MCDM factor design

have any land. The producer has the technology to produce bio-gas from manure and have income from that.

The main *institutional statements* are derived from the following laws and regulations in this system:

- Manure policy: The *fertiliser law*, the *animal production rights* and *EU milk quota regime* are the main rules related to manure. The fertiliser law indicates how much nitrogen and phosphate a farmer is permitted to use for fertilising with manure. Based on type of land and the number of hectares, farmers calculate the amount of nitrogen and phosphate they need for spreading manure on the land. We assume that all cattle farmer agents perform *derogation*. Therefore, they are allowed to use more kilogrammes of nitrogen per hectare. However, this is only permitted if the nitrogen originates from cattle manure.
- Farmer-farmer transport: If cattle farmer agents spread manure equivalent to 80% of their phosphates allowance, they are permitted to use *Farmer-farmer transport*. This rule allows a cattle farmer to transport the manure to neighbouring farms (within the same area) against reduced costs.
- **Subsidy**: An agent who takes the role of a producer receives subsidy for his supply of biogas.
- Abandonment and expansion of a farm: Farmers are confronted with policies such as the *ammonia action programme* and the *animal welfare norms* which mainly affect pig farmers because they use animal housing systems. A considerable percentage of the farmers are expected to abandon their farm by 2013. On the other hand, both cattle and pig farmers have a tendency towards up scaling. Depending on the individual circumstances (e.g., scale of the farm and presence of a successor), an agent might decide to abandon or expand. In case an agent decides to expand, he is allowed to increase the number of animals and hectares<sup>11</sup>. An agent is permitted to expand his farm only twice in the simulation.

These policies are structured into ADICO statements in Table 4.9.

#### Physical structure

We define the following *physical components*:

 $<sup>^{11}</sup>$ An agent is only allowed to increase the number of hectares if land is made available to him by an abandoning neighbour within the same location.

Name	Attributes	Deontic Type	aIm	Condition	Or else	Type
Manure	Landowner	Prohibition	Use more than 170 kilo-	Spreading manure	1	rule
policy (in-			grammes of nitrogen per	on land		
cluding			hectare (in case <i>derogation</i>			
derogation)			applies this amount is in-			
			creased to 250)			
Farmer-	Animal (Cattle)	Permission	Transport their excess	Have cattle and are	I	rule
farmer	Farmer Agent		to neighbouring (cattle)	able to use more		
transport			farmer agents within the	then $80\%$ of the		
			same location	phosphate for ma-		
				nuring of own land		
Subsidy	Producer	Permission	obtain a basic amount	If invest in techno-	I	rule
			granted by subsidy	logy		
Abandon of	Animal (cattle and	Obligation	Reduce the ammonia	If has animal hous-	Ι	rule
farm	pig) farmer agent		emissions from animal	ing		
			housing systems at farm			
			by 2013			
Expansion of	Animal (cattle and	Permission	Expand their farms	Only two times	I	rule
farm	pig) farmer agent					

Table 4.9 – The Institution table for the bio-gas energy system

- Animal: Animal represents all the animals in a farm. We assume that no farmer can have both cattle and pig. Therefore, 'animal Type' can be 'pig' or 'cattle' for each farmer. Another property of animal is 'number' of animals. The behaviour of the animal is 'produce manure'.
- Land: The 'type of land' property can be 'crop' or 'grass'. Land also has the *property* 'size' in hectares.
- **Manure**: All animals produce manure. Depending on the type of animal, an estimation is made of the annual production and mineral composition. Manure can be processed (i.e., an *affordance*).
- **Digestate**: Digestate is a bi-product of bio-gas with similar composition to manure. Since manure from cattle differs in composition compared to manure from pigs, digestate from cattle also differs from pig digestate. Digestate can be produced (i.e., an *affordance*).
- Artificial fertiliser: Artificial fertilisers are used by farmers because they are limited in the amount of manure they can spread on land. Artificial fertilizers can be 'spread' on land(i.e., an *affordance*).
- **Bio-gas**: The volume (m<sup>3</sup>) of bio-gas depends on the volume of manure that is processed. The *affordance* of bio-gas is 'produced'.

# Group Formation in MAIA

A group is a social entity that is composed of actors who share information and resources. Groups can be defined as agents in MAIA (See for example Case Study 3). However, groups can also be emergent structures in an agentbased model. In MAIA, an agent makes a decision about joining another agent, considering the added benefits of this coalition. As soon as two agents join each other, a group is formed with a unique identification. Other agents then make decisions to join this group or look for other agents to make a new group.

# **Operational structure**

We defined four *action situations* in the bio-gas production arena:

- **Farming**: Manure is produced according to the number of animals available on each farm. Farmers buy artificial fertilisers and the artificial suppliers sell the fertilisers. Farmers spread the manure on their lands. They may also expand or abandon their farms.
- Manure distribution: Farmers request for distribution, intermediaries accept manure and digestate. Landowners request for manure Intermediaries distribute manure to different farms based on the information they have about the need of the farms.

	Update technology costs	Manure production	Invest in bio-gas	Cooperate in investment	Produce bio-gas	Pay technology costs	Abandon	Expand	Fertilize land	Distribute manure to neighbours	Calculate manure acceptance price	Send manure to intermediary	Distribute manure (intermediary)	Buy artificial fertiliser
Manure (cattle/pig)	i	d	d	d	d	i	i	i	d	d	d	d	d	i
Acceptance price manure	i	i	d	d	i	i	i	i	i	d	d	d	d	i
Money of farmers	i	i	i	i	d	d	i	i	d	d	i	d	d	d
Number of farmers	i	i	i	i	i	i	d	i	i	i	i	i	i	i
Number of animals	i	i	i	i	i	i	i	d	i	i	i	i	i	i
Technology costs	$\mathbf{d}$	i	i	i	i	$\mathbf{d}$	i	i	i	i	i	i	i	i
Green gas volume	i	d	i	i	d	i	i	i	i	i	i	i	i	i
Number of producers	i	i	d	d	i	i	i	i	i	i	i	i	i	i
Number of groups	i	i	d	i	i	i	i	i	i	i	i	i	i	i
Size of groups	i	i	d	d	i	i	d	i	i	i	i	i	i	i
Composition of groups	i	i	$\mathbf{d}$	d	i	i	i	i	i	i	i	i	i	i

Table 4.10 – Scope matrix

- **Bio-production**: The farmers may invest in the technology and produce biogas. The annual technology costs are paid yearly in this action situation.
- **Bio-production with cooperation**: Farmers may form groups to invest in the technology. They will produce bio-gas collectively. Agents consider individual investment before considering cooperation. Both cattle farmer agents and pig farmer agents can cooperate and form groups as long as they are situated within the same location.

We incorporate the effect of 'economy of scale' by making the annual technology costs dependent on the production capacity. In other words, above a certain production capacity, these costs will become a function of the number of agents within the group.

# **Evaluative structure**

The scope and validation matrices are illustrated in Tables 4.10 and 4.11.

# 4.5.3 Simulation Implementation

The structure of this simulation was very similar to the previous case study. We initialized parameters using available data sets and implemented a time loop that

	Update technology costs	Manure production	Green gas investment	Cooperate in investment	Green gas production	Paying technology costs	Decision to abandon	Decision to expand	Manuring land	Distribute manure to neighbours	Calculation of manure acceptance price	Distribute manure to intermediary	Distribution of manure by intermediary	Buy artificial fertiliser
Manure (cattle/pig)	i	d	d	d	d	i	i	i	d	d	d	d	d	i
Green gas volume	i	i	i	i	d	i	i	i	i	i	i	i	i	i
Acceptance price manure	i	i	$\mathbf{d}$	$\mathbf{d}$	i	i	i	i	i	d	d	d	d	i
Money	i	i	i	i	d	$\mathbf{d}$	i	i	$\mathbf{d}$	d	i	d	d	d
No. of farmers	i	i	i	i	i	i	d	i	i	i	i	i	i	i
No. of animals	i	i	i	i	i	i	i	d	i	i	i	i	i	i
Technology costs	d	i	i	i	i	d	i	i	i	i	i	i	i	i

Table 4.11 – Validation matrix

repeated the action situations 50 times. The agents randomly entered the action arena in each tick and executed actions that they were allowed to perform. Since we had three different experiments, in the time loop the action situations were only activated if they were part of the current experiment. In the first experiment, no bio-gas technology existed. Therefore, agents only took part in farming activities and the manure market. In the second experiment the agent had the possibility to invest in bio-gas technology. In the final experiment, the agents took part in all the action situations and were also able to cooperate in bio-gas investment and production. Data was stored and analysed, similar to the previous case study.

## 4.5.4 Simulation Results

Three different experiments were set out to achieve these targets: (1) gain insight into the dynamics of the manure distribution system, (2) explore the potential for manure-based energy production by individual farms, and (3) explore the potential for farmer cooperation for manure-based energy production.

Because of the financial problems of this sector and the continuous change in institutional settings, the number of pig and cattle farms decrease over time especially the small scale farms. Pig farmers are more vulnerable to abandonment because of the animal welfare rules and ammonia action programme that only address an animal housing system. However, the number of animals stays the same over time since the bigger farms take over the animals of the closed farms. Furthermore, the manure collected by intermediary agents each year increases over time which shows an increase in the problem of manure distribution for farmers.

In the second experiment, we observe that only a minority of farmers invest in bio-gas production. This has several reasons including (1) insufficient manure
or capital, (2) unclear long term benefits of this investment and (3) institutional pressure on farmers, making them reluctant to think of bio-gas production as a long-term strategy. The amount of bio-gas produced over 15 years in this experiment is illustrated in the two diagrams in Figure 4.7. The circled area indicates the volume of bio-gas produced for annual technology costs of  $\in$ 80,000 (actual estimation) and a revenue of  $\notin$ 1,04 per  $Nm^3$ (current maximum revenue).

During the interviews, the farmers were not enthusiastic about cooperation. However, the third simulation shows that cooperation benefits both farmers and the Overijssel project in the long run. Figure 4.8 depicts the total volume (from cattle and pig) of bio-gas produced, which is relatively high compared to individual production in Figure 4.7. While in our previous experiment only those farmers who had large-scale farms and high capital invested in the technology, by cooperating, smaller scale farmers with less capital also had the chance to invest.

#### 4.5.5 Lessons Learnt

The main goal of this case study was to find out how qualitative and quantitative data about a system can be incorporated into an agent-based model using MAIA.

A MAIA model, with the current version of the MAIA tool, does not capture quantitative data. However, we loaded the data into the model during implementation to instantiate agents and initiate the simulation. The qualitative data (result of interviews) were successfully used to build the MAIA model. The case study also verified the possibility of structuring the formal regulations into ADICO statements. Finally, after performing this case study, there were no more conceptual changes to the MAIA meta-model which made us conclude that the meta-model may have reached a stable state for describing the type of systems we are interested in.

**Redundant Concepts** During the implementation of this case study we managed to extract the general rules for implementing a simulation from a MAIA model. Therefore, the redundancy was not really about the concepts but rather about the implementation process we had defined while conducting our four case studies. For example, we decided that we do not need to implement actions for every agent but rather define one that all agents can enter to perform and exit. We decided that the concept of institution and role can also be shared in the simulation among different agents and we do not need to define them separately for each agent. This implementation choice is in line with the concept of institutional structure which is rather a shared and general structure that affects agent behaviour rather than internal to the agents.

**Usefulness and Usability** Using MAIA to develop the bio-gas system proved to be both effective and efficient. The modeller was able to capture a known system with available data efficiently because MAIA was a template that she could structure the information into. Furthermore, since the modeller and programmer were different people, cooperation was highly facilitated with MAIA.



Figure 4.7 – Green gas production without cooperation (De Korte, 2012)



Figure 4.8 – Green gas production with cooperation (De Korte, 2012)

The modeller had some difficulty in learning and understanding the concepts because they were relatively new to her. However, she became skilful in using MAIA during the project.

# 4.6 User Survey Evaluation

In this chapter, we briefly summarized four case studies which were conducted using MAIA. We presented our reflections on each case study in the 'Lessons Learnt' sections. To cover a range of viewpoints and avoid biased evaluation, we distributed a questionnaire among the users of MAIA to learn from their feedbacks and experience from using MAIA.

The questionnaire which can be found in Appendix A, has two parts: a set of questions asking to grade from 1 to 7 (1 for fully disagree, and 7 for fully agree) on Likert-type scale and a set of open questions. The first set included questions on the usefulness and usability of MAIA, based on the indicators defined in Section 4.1. In the second set, we asked the MAIA users to give the benefits and drawbacks of using MAIA, name the difficulties they experienced and recommend areas for improvement.

Overall, the usability and usefulness of MAIA for the four case studies can be seen in Table 4.12. In total we received 5 responses<sup>12</sup>, excluding the developers of MAIA. On average, the users gave a grade of 3.3 (average) to ease of understanding the concepts and learning to use MAIA. They all agreed that MAIA is helpful as a participatory tool giving a grade of 4.3. Furthermore, all the users were extremely optimistic about the application of MAIA to other problems in their domain giving

 $<sup>^{12}\</sup>mathrm{We}$  only considered those users who actually conceptualized a complete agent-based model with MAIA.

		Usefi	ılness			Usa	ability	
Case Study	Effectiveness	Cooperation	Applicability	Efficiency	Ease-of-use	Learnability	Ease of becoming skilful	Clarity/ Understandability
Case Study 1	6	-	6	6	-	-	-	-
Case Study 2	5	3	6	5	4	2	5	3
Case Study 3	6	4	7	3	6	5	5	5
Case Study 4	5	6	6	4	6	3	6	2

Table 4.12 – Usefulness of MAIA for developing the four case studies and its usability.

it on average the grade 6.25.

The benefits of using MAIA according to these users include: assistance in structuring the model which they had greatly missed in their previous experiments with ABMS, helping to identify model elements, feasibility of reaching an ABM from high level concepts, facilitating discussions with domain experts, helping to develop more complex, yet manageable models.

As one of the users indicated: "MAIA *allows* a researcher or modeller to use it in the best possible way. With the flexibility of MAIA, the quality of the model is more defined by the capabilities of the researcher than the framework". Another user explained that "The fact that the MAIA framework is comprised of a divers range of relational *structures* encouraged us to explore different system components and forced us to truly evaluate their relevance for considering in the ABM".

The respondents of the questionnaire also had remarks on the drawbacks of MAIA. They indicated that a considerable amount of background knowledge on social systems and institutions is necessary for using MAIA. They were also concerned about the time required to use MAIA due to the iterated relations between the tables and diagrams. We addressed this problem by developing a tool which automatically manages these relations. Another concern about MAIA was that a modeller believed that there were insufficient details at some points during the implementation process. We addressed this problem by keeping our abstract level but providing a more detailed, formal structure which will be presented in Chapter 5 of this thesis. Finally, one of the respondents suggested that having a library of best practices and common algorithms (e.g., negotiation) would facilitate the use and understanding of MAIA.

One other question addressed the difficulties the modellers faced when they were using MAIA. Their response indicated that the separation of roles and agents was one of the most difficult parts of using MAIA. This is why in many of our case studies, the modellers defined separate roles for each agent type (i.e., one-to-one relationship between roles and agents) and avoided many-to-many relationships between agents and roles. This choice does not limit MAIA in any way and is an easy approach to overcome the difficulty for beginners. Another respondent found institutions difficult to model but we addressed this issue by introducing the ADICO structure, which was not present in our first version of MAIA.

A point to mention here is that, there are two different views on institutions (Hodgson, 1988) that may cause confusion for the users of MAIA: an organization view where many aspects of the system including actors are considered as the institution (Esteva et al., 2001) and a system of established rules view (Hodgson, 1988), which we are following in this research. Therefore, in our tutorials we must be clear about which terminology of institutions we are using.

Overall, we received many positive feedbacks and encouragements from the users of MAIA. Nonetheless, we strived to address most of the drawbacks of MAIA and the difficulties it may give to its users. There are still improvements to be made which will be reflected upon in the final chapter of this manuscript.

# 4.7 Conclusion

In this chapter we explained the four case studies conducted to evaluate MAIA in terms of useability and usefulness. During the modelling process of these case studies, we tried to make MAIA more parsimonious by excluding redundant concepts. To explore the additional benefits of MAIA, two case studies were redeveloped. Finally, to broaden the scope of our evaluation, feedbacks from the users of MAIA were also incorporated.

In general, as ABMS is used to address complex situations, the need for various expertise in the modelling process calls for participation. Therefore, the use of MAIA as a participatory tool is a major benefit that should be taken for granted. Since there are many different ways of describing the social (or the technical) context of agent-based models, communication is served if a common framework like MAIA is used. For the modeller, some of the assumptions she makes on the structure of society may seem trivial, but they may not be, in a broader context and thus will need to be explained, preferably in a standard way. Furthermore, the MAIA metamodel encourages the modeller to be explicit about the elements she includes and the ones she excludes. This may be too much effort for abstract models that focus on one or two elements, but for simulations of complex socio-technical systems such as the ones we studied, it has been certainly worth while.

The limitations that we confronted during the case studies, gave us guidelines on how to further continue this research. We already presented the web-tool in Chapter 3, which diminishes the difficulties regarding the interrelations between the concepts in MAIA. To ease the learning process of MAIA in terms of usage and social concepts, we will provide detailed tutorials and definitions which we have already started by building the MAIA website: http://maia.tudelft.nl. Finally, to provide more details on the concepts that would actually lead to programming code, possibly automatic, we need more formalization of MAIA concepts. This will be the topic of the third part of this thesis. However, we would still like to stress that, the MAIA meta-model at its current level of details can still be applied as a useful ABMS tool in various aspects as we saw in the case studies.

# Part III

# FORMALIZED INSTRUMENT FOR AGENT-BASED MODELLING

Do not go where the path may lead, go instead where there is no path and leave a trail.

Ralph Waldo Emerson

# 5 Model-driven Development for MAIA

# 5.1 Introduction

A simulation imitates the operation of a system over time, to show how it evolves (Banks et al., 2000). In an agent-based simulation, agents are the active entities who are scheduled to perform operations in a given space (Pavon et al., 2008). The simulation is the result of running a model that describes the agents, actions and space. Although modelling and simulation are recognized as two different stages of ABMS, there is no consensus about the process of making a running model (i.e., simulation) (Pavon et al., 2008).

For some developers, there is no transition between a model and a simulation because they describe the model in the low-level programming language which produces the simulation. However, low-level programming languages are difficult to use especially for non-programmers (Ramanath and Gilbert, 2004). Furthermore, while this simulation approach might be feasible to take for smaller simulations, it becomes more difficult and even impossible as the complexity of the simulations grow.

In other cases, the modeller makes a description of the model in pseudo code (e.g., (Frank et al., 2011)), diagrams (e.g., (Okada, 2011; Chappin, 2011; Behdani, 2012)), equations (e.g., (van der Veen et al., 2012; Abbasy et al., 2011)) or a generic modelling language such as UML (e.g., (Bagni et al., 2002)) and translates the model into a simulation in an ad-hoc manner. This approach however, does not support participatory simulation because there is no common or standard language between the modellers. Furthermore, since there are no predefined concepts and relations, redevelopment of the simulation or re-use of its components is difficult.

Unlike the first approach, MAIA is a high level language and therefore, requires translation to a simulation. Yet, MAIA manages the complexity of a model by structuring it into the key characteristics and behaviours of the system it represents. Unlike the second approach, the structure of the description of a model is predefined in MAIA. Therefore, it can be used as a common language to support participatory simulation development. Furthermore, because there are predefined components for a model, the re-use of the component and redevelopment of the simulation become more feasible. However, to produce running simulation, from MAIA models, more detailed specification about the transition is required.

The transformation from a high-level modelling language such as MAIA to a simulation is in line with Model-driven Software Development (MDSD) practices. MDSD, as explained in Chapter 2, facilitates the development of a software system from a conceptual model. The major requirements for MDSD are meta-models (e.g., MAIA) to describe 'what' to model, rules to show 'how' to make software, and transformation platforms to generate software.

The goal of Part III of this thesis is to further formalize MAIA within a MDSD process. In this chapter, we explain *how* a MAIA model can be translated to an executable simulation. In the following chapter, we provide formal syntax and semantics for MAIA so that it can be parsed by the computer and used for automatic code generation.

The remainder of this chapter is structured as follows. In Section 5.2, we define an agent-based simulation architecture and we also introduce a running example which we will use throughout the chapter to explain the process of making a simulation from a MAIA model. In Section 5.3, we explain the general concept behind a MDSD approach and show different ways of achieving simulation code from a MAIA model. In Section 5.4, we explain the direct approach of getting to running simulations in object-oriented programming languages from MAIA. In Section 5.5, we explain the mapping process between MAIA and a platform independent modelling language which can then translate MAIA models into various platforms including Repast. In Section 5.6, we conclude this chapter.

# 5.2 Agent-based Simulation Architecture

Once a model is conceptualized, the next step is to perform the execution of the model or in other words simulate it. Simulation means animating the specification of the system over time to see how it evolves (Pavon et al., 2008). Simulation platforms (e.g., Repast, Netlogo, Mason) include several shared modules illustrated in a class diagram in Figure 5.1:

- The agent: The active entity that is responsible for performing actions and communicating with other agents and/or the environment.
- The agent list: To keep track of all entities, agents are normally stored in sets.
- The scheduling mechanism: The definition of the period a simulation runs. There are two scheduling mechanisms that a simulation can follow: time-driven

and event-driven (Pavon et al., 2008). In time-driven scheduling, agents get turns to perform actions in time steps. In event-driven scheduling, a central coordinating mechanism synchronizes the agents.

- The space: The spatial or logical representation for agents and other components in the system.
- The main model: Where all the entities are gathered. It is also in charge of the context of the simulation experiment (e.g., initialization of variables, and visualization and analysis of result) (Pavon et al., 2008).

In the following sections, we will use this architecture to transform a MAIA model into an executable simulation.



Figure 5.1 – An agent-based simulation architecture.

#### Working Example

Similar to the previous part of this thesis, we will use an example to explain concepts in a more tangible manner in this part of the thesis.

This example is about method of care for elderly people. In this setting, the government wants to create incentives for people to take care of their own elderly family members instead of sending them to care homes or hiring nurses (Mantelzorg, 2013). The example explains a social setting where people decide how their elderlies can be taken care of. Factors such as means of transport, the health situation of the parent and price of care homes affect the decision of the individuals. The modelling goal is to find out which of the following options can be more effective in this situation: 1) tax return to those people who take care of their elderly families, 2) extra free days and flexible working days or 3) higher taxes on care homes.

## 5.2.1 A MAIA Simulation

Using the simulation architecture, we map the concepts and relations in MAIA into a simulation<sup>1</sup>.

Table 5.1 roughly shows where each MAIA concept fits in the simulation architecture. In the following, we will explain how each of these modules function in more detail.

Table 5.1 – Mapping between MAIA concepts and simulation modules.

Simulation Element	Related MAIA Concepts
Agent	The agent concept in MAIA represents agents
	in the simulation. The agent is extended with
	the responsibilities and objectives of the po-
	tential roles he may enact. Also, the physical
	components that an agent may possess are in-
	cluded in the definition of the agent.
Agent List	For each MAIA agent type, we define a list.
	All public physical components are also stored
	in lists.
Scheduling Mechanism	This is not present in the MAIA meta-model
	and would need to be added to the simulation.
Space	The action arena, institutions and physical
	connections provide the operational, institu-
	tional and physical spaces in the simulation
	respectively.
Main Model	Initialization is not provided by MAIA. How-
	ever, the evaluative structure of MAIA which
	addresses the goal of the simulation with a set
	of variables, builds the result section of this
	element.

Table 5.1 – Mapping between MATA concepts and simulation module

#### Agents

An agent in a MAIA simulation is a *decision making* entity who performs actions in the action arena. He has information about which roles he may be enacting and the physical components he owns or may have access to. Agents will perform *actions* based on the roles they enact, the institutions they must comply with and the physical components they may possess.

Agents Decision Making MAIA agents make decisions about performing actions. For each action, the agents decide whether to perform that action or not (Scharpf, 1997). However, they can also follow the basic procedure in simulations

 $<sup>^{1}</sup>$ Even though our simulation implementation is in Java, we keep this explanation abstract and syntax free so that it would be usable for other conversions. Later on, we will also briefly present the Java specific implementation details.

and execute actions intuitively. The default decision making algorithm in MAIA is Multi-Criteria Decision Making (MCDM) (Guitouni and Martel, 1998) presented as pseudocode in Listing 3. This algorithm can be replaced by any other mechanism design algorithm in MAS. In each decision making routine, the agent *may* check a number of conditions. In the family example, he checks whether the parent has been in a care home before and if the parent's age is above a certain value. The agent also has a number of decision aspects (i.e., parameters the effect a decision) with numerical values which are prioritized with weights. These aspects can be properties of the agent, other agents, physical components or global variables among others. If the sanction has a numerical value (e.g., fine), the agent may also take this value into account. If the weighted sum of aspects is bigger than a specified threshold and the conditions hold, the agent decides to perform an action.

	BEGIN MCDM(agent)
2	IF conditions
	weightedSum = Aspect1 $*$ Weight1 + + AspectN $*$ weightN
4	IF weightedSum > decision-threshold
	RETURN TRUE
6	END IF
	END IF
8	RETURN FALSE
	END MCDM

Listing 3 – Multi-criteria decision making algorithm.

Unlike MCDM aspects that need to have numerical values, decision conditions are more flexible in defining decision influencing factors. Nonetheless, the condition part of a decision and the aspects can be used interchangeably, if the conditions have numerical values (e.g., the age of parent can be considered as a condition (age > 60) or an aspect  $(age \times weight)$ ).

In each decision making procedure, priority is given to an institution with sanction (i.e., rule) (rule >> norm >> shared strategy). So there is, by default, a higher chance of obeying an institution that has a sanction. The 'willingness to comply' attribute, given as a predefined property to every agent, affects the decision when there is an institution associated to the action the agent is going to perform. For example, if a norm tells him to visit his parent every day, his 'willingness to comply' value among other factors, will influence the outcome of the decision about visiting his elderly parent.

**Agents in Action** In the current implementation, actions are static descriptions that agents enter, execute and exit. When an agent wants to execute an action he first checks the preconditions for performing that action. If these conditions are met, he will proceed differently according to the type of action.

• In its simplest form, the agent executes the action by only checking the preconditions (e.g., If agent.isAlive & agent.isElderly() then requestHelp()). This type of execution is used if the body of that action is an intrinsic capability of an agent (e.g., eat()) or a behaviour of physical component (e.g., getFlatTires()).

- The agent may go through a decision making process before executing an action. If the preconditions to execute an action hold, the agent performs the action only if the result of his decision making process is also positive. The body for this type of action, can be an institutional responsibility of a role (e.g., agent.child.takeCareofParent()), or an intrinsic capability of an agent (e.g., eat()). Note that in the precondition of the action with institutional responsibility, it is checked whether the agent has the associated role (e.g., if agent.isParent()).
- An action may also be related to an institutional setting and therefore have an ADICO statement attached to it. In that case, depending on the type of institution (rule, norm or shared strategy), different action processes take place. With this form of action execution, there is always a decision making process. This is because MAIA agents have the freedom to choose whether to comply with an institution, weighing their own personal preferences against what an institution is putting forward to them. The body of this form of action is always an institutional responsibility. We will explain this type of action in more detail next.

There are three types of institutional actions: rule-based actions, norm-based actions and actions based on shared strategy. Listing 4, presents a rule-based action execution procedure. If the preconditions for an action to execute hold, and the institution conditions also hold, the agent makes a decision about executing or not executing that action. For example, when an agent is in the 'pay care home fees' action, he checks the precondition (parent.inCareHome), then he checks the institution condition (isEndOfMonth). The agent then *decides* about paying the care home or not.

If an agent executes an action that he is prohibited to perform (or does not execute one that he is obliged to perform), a *flag* is raised for that agent automatically without any third party observer. For example, if an agent decides not to pay for the care home, the flag for the sanction which indicates that the elderly must move out is raised. This sanction is an institutional statement by itself (The care home is permitted to *move an elderly person out* if the fees are not paid).

This flag will stay on, until the rule enforcer agent (i.e., the care home) checks the status of the agent. The rule enforcer may come across the agent in the same tick (all agents get the chance to perform an action in each tick) or much later. It will only pose a sanction on the agent (e.g., move his elderly out), if the conditions for that sanction still hold at the time he is inspecting that agent (paymentStatus = notPaid). He will turn the agents flag off, but may only pose the sanction based on the current status of that agent.

A norm-based action and an action based on shared strategy follow similar procedures except that there is *no sanction* for a norm-based action and *no deontic or sanction* for a shared strategy.

	BEGIN EXECUTE(agent)
2	
	IF precondition holds
4	IF institution-condition holds
	IF agent-decision(action) AND institution-type = PERMISSION OR OBLIGATION
6	execute action
	apply post-condition
8	END IF
	IF $agent-decision$ (action) AND institution-type = PROHIBITION
10	execute action
	apply post-condition
12	execute sanction
	update objective
14	END IF
	IF NOT $agent-decision(action)$ AND $institution-type = PROHIBITION$
16	update objective
	END IF
18	IF NOT $agent-decision(action)$ AND $institution-type = PERMISSION$
	no action
20	END IF
	IF NOT $agent-decision(action)$ AND $institution-type = OBLIGATION$
22	perform sanction
	update objective
24	END IF
	IF institution—condition does not hold
26	IF agent-decision(action)
	execute action
28	apply post-condition
	END IF
30	END IF
	IF precondition does not hold
32	no action
	END IF
34	END EXECUTE

DEGIN EVECTORE

. \

Listing 4 – Executing a rule-based action.

When an agent performs an action that has a rule associated to it, the consequence of non-compliance is implemented as explained above. However, non-compliance to norms and shared strategies are only reflected in the status of the agents (e.g., agent.parent.notHelped) or global variables in the system (e.g., numberOfPeople-InCareHome = all elderly agents in the simulation).

An action can have more than one associated institution. If there is at least one rule in the list of institutions for an action, a rule-based procedure is executed and the conditions of all institutions are checked. If there is no rule in the set of institutions for an action, but at least one norm, the norm-based procedure is executed. Otherwise, if there is a list of shared strategies, the agents follow a shared strategy procedure.

As we will explain in Section 5.2.1, actions are ordered in plans. An entity action is itself defined as an atomic plan. Institutions can also apply to a non-atomic plan which contains a set of entity actions. Nonetheless, the agent follows the same procedure for considering institutions. In MAIA, the institutions associated to a role are in-line with the objectives of that role. By default, for every role that the agent enacts, there is an attribute called 'objective distance' (default value: 5) that reflects how far the agent is from reaching that particular objective. The agent's compliance with institutions affects this attribute (increases or decreases the value). In the current implementation, when agents disobey norms, the objective of the agent is updated (i.e., the 'objective distance' is increased) to reflect the idea that non-compliance to institutions takes agents further away from meeting the objective of the role they are enacting.

At the beginning of each tick, the agents check their 'objective distances'. If the value is above 10, the agent increases his 'willingness to comply' which is another default attribute of the agent. An agent's 'willingness to comply' value (default value: normal distribution: [0-10]) *partly* determines how much he complies with institutions in his decision makings as we explained above.

#### Scheduling Mechanism

Many ABMS platforms are time-driven (e.g., Repast, Netlogo, Swarm). For example, in Repast, the time-activated agents perform a predefined list of actions. Time-driven scheduling is easier than designing a central coordinator. However, event-driven simulations are more flexible in the order of actions and therefore, there may be more possibilities for emergent patterns and structures in simulations (Pavon et al., 2008). Nonetheless, while emergent patterns are desirable for ABMs, in the case of simulation, modellers also need to have control over the artificial society to track what is happening in each time step.

As suggested by Pavon et al. (2008), we combine both mechanisms (i.e., timedriven and event-driven) for our scheduling system. In each time step, each agent is activated to enter the action arena which has a pre-specified order of actions. However, the agents have the freedom to decide to execute (or not) the action they come across. Therefore, there is more diversity in the order of actions because agents do not necessarily have to perform every action available to them, and unanticipated behaviour can still emerge. We must emphasize that the meaning of time is not equivalent to 'physical' time and only used to give turns to agents.

The other task of the simulation mechanism is to organize the agents by systematically randomizing their order of activation in each time step. Finally, the scheduler sets the duration of the simulation (i.e., the number of times the action arena repeats).

#### The Space

Compared to other simulation platforms, the space module of a MAIA simulation is relatively complex. It contains the operational, the physical and the institutional spaces all linked to each other as depicted in Figure 5.2. The main component of space that relates concepts in the different space modules is the action arena.

Action Arena Figure 5.3, shows how agent instances enter the action arena one by one in each tick. Each agent instance may *own* a physical component (e.g., car)



Figure 5.2 – The space module in a MAIA simulation forms around the action arena.

as he enters the action arena (illustrated by a square next to the agent).

When an agent enters the action arena, there is a set of actions he may be able to perform. The order of these actions is specified by the modeller with the plan concept in MAIA as explained in Chapter 3. The agent searches the action arena by going through the actions one by one according to the plans. Plans and the steps in each plan are uniquely identified for each agent instance. At the beginning of the simulation, the plan id and step id of each agent point to the beginning of the action arena: i.e, the first action in the first action situation.

As soon as the agent finds an action that he *may* perform (i.e., preconditions meet), he either executes the action or decides (as explained) not to perform that action. He will then exit the action arena even if he has decided not to perform the action. The agent will update his step id and plan id to point to the next action for the next tick.

In the current implementation, actions are not nested. If, by performing an action, the agent is triggering another action, this is implemented by the agent raising a flag which will result in other agents or himself performing an action that is the consequent result of the current action. However, the action that is to be executed may not happen in the same time step. I.e., the execution of the action will happen in the next time step if the performer is the same agent because agents can only perform at most one action per time step. Nonetheless, if the performer is another agent who has not taken turn yet, the action may still execute at the same time step. We would like to stress here again, that time steps do not necessarily represent real time (e.g., year, month) and are defined to allow "the execution of one action per agent".

Another point to mention regarding the order of actions is that currently, agents cannot decide between a list of actions because the result of a decision is always a



Figure 5.3 – Simulated environment using MAIA. The dotted circles represent the plans within each action situation. An arrow between two plans shows a sequence. The arrow with circles on both ends represents an alternative.

boolean value (I do this or I don't). Therefore, if an agent wants to choose an action from a list of possible actions, he has to weigh each option separately. In the family care example, we want the agent to choose between (1) send to care home, (2) hire nurse, and (3) self care for the elderly. In the current implementation, each of these are separate actions with separate decisions. However, when the agent is deciding whether to put his elderly parent in a care home, he also considers the cost of hiring nurse or tax return benefits while making the decision on care home. In other words, even though we have a boolean outcome for each decision, we may still implicitly consider other actions in one decision.

**Institutional Space** The institutional space is intertwined with the operational space in the sense that the definition of ADICO statements is programmed into actions. As explained in Section 5.2.1, agents check the institutional condition for every entity action (or plan) they enter. For every non-compliance to a rule, a flag is raised in the institutional space to indicate to the rule enforcer that an agent should face a sanction.

**Physical Space** The physical space has two aspects. It is a place-holder for public physical components and it also captures the visual space in the simulation.

The *public* physical components are shared between all agents in the model. There is only one representation of each public physical component (e.g., train) in the default simulation. The status of these components is updated according to the actions agents perform on them. If a simulation requires a visual representation, each node (i.e., physical component) must have coordinates as its properties. The physical connections between physical components provide the edges of the network if required. To make visual representation for agents, a physical component (e.g., body) needs to be defined for the agent and given to him during conceptualization.

#### The Main Model

The main model initializes the variables in the simulation including the agents, physical components and other global variables. It uses the scheduler to order and activate agents to enter the action arena one by one. Finally, it uses the evaluative structure of MAIA to define resulting variables that can be visualized in relation to the variables in the simulation. The main model is illustrated in the class diagram in Figure 5.4.



Figure 5.4 – The main model in a MAIA simulation.

A house keeping procedure at the beginning of each tick performs several routine tasks. For example, it assigns roles to agents and each agent gets the chance to check his objectiveDistance for every role he is enacting and update his willingnessToComply according to this value.

**Evaluation of Results** The third and final component of the main model is the analysis component. The variables defined in the evaluative structure of the MAIA model are calculated in every tick. They are stored in separate files per tick and also per independent variable. In the MAIA model of the family example, we defined problem domain variables as: numPeopleInCareHome, numPeopleWithNurse, numPeopleFamilyCare. We monitored these variables for different values of the independent variables: tax return, free days and tax on care homes. Figure 5.5 shows some results from the family care example. In these diagrams, we can see that increasing the salary of the people or in other word, tax return, would be the most effective incentive for them to take care of their elderly parent. However, hiring nurses would also be increasing because of the extra income people receive. Therefore, the option of giving free days to people who take care of their families seems to be more effective if the strategy is to also lower the number of nurses. The analysis of such results is also aided by the relations between variables and entity actions specified during conceptualization in the evaluative structure.

In the description of a simulation given above, we tried to abstract away from any specific platform or programming language (e.g., Java and Object-oriented programming) so that other simulation platforms could also be used with this description in order to produce simulations from a MAIA model. In the next sections, we will explain how this description can be used to make simulations using a model-driven procedure.



Figure 5.5 – Simulation results from the Family Care example comparing different policies.

# 5.3 A Model-driven Approach to Build Simulations

The main requirements for a MDSD approach as explained in Chapter 2 are (1) metamodels that describe domain specific models, (2) transformation modules that map one type of model to another (e.g., MAIA model to Java Code) and (3) platforms that produce executable software.

A basic model-driven approach is illustrated in Figure 5.6. A *computational in*dependent model (CIM) describes the context of the model at a conceptual level. It focuses on the requirements of the software systems (i.e., simulation in this context) (OMG, 2013). According to this description, since MAIA is designed to address 'what' to put in a simulation and does not give low-level programming details, this framework can be considered as a CIM *meta-model*. An agent-based model conceptualized using MAIA (e.g., e-waste, consumer lighting) is a CIM instance (i.e., domain model) of the MAIA meta-model.



Figure 5.6 – Model-driven Software Development

A platform specific model (PSM) describes the realization of the conceptual model for a specific software platform<sup>2</sup>. Likewise, a PSM is an instance of a PSM metamodel. The transformation between a CIM and a PSM, requires mapping modules that describe what each concept in the CIM is translated to, in a PSM.

If we assume the specifications of the Java programming language (e.g., class, object, method, attribute) as a PSM meta-model, the minimum requirement to get to executable code (in Java) from a MAIA model is to define a mapping module that performs the transformation<sup>3</sup>. After developing all of our case studies in Java, we extracted common protocols for transforming MAIA models to simulations in Java. We will explain this procedure of model-to-code transformations in Section 5.4.

There are also other options for transforming a CIM into executable code. For example, taking the Repast meta-model as a PSM meta-model, we can translate

 $<sup>^2</sup>$  A platform can be considered as a programming language (e.g., Java), programming platform (e.g., Eclipse for Java), or a simulation platform such as Repast.

<sup>&</sup>lt;sup>3</sup>Because it is possible to define many CIMs and PSMs at different levels in the MDSD process, we can even consider the Java Language as a PSM meta-model. However, one may also argue that the conversion between MAIA models and Java Simulations is direct transformation to code (i.e., CIM to code). This is still a recognized model-driven approach as described by The Object Management Group (OMG, 2013).

MAIA to a model for the Repast platform. However, we can also go one step further in facilitating model-to-code translation by introducing a *platform independent model* (PIM) into the MAIA development process. A PIM focuses on the operation of a software system while hiding the details necessary for a particular platform (OMG, 2013). As illustrated in Figure 5.7, a PIM acts as an interface between a CIM and a PSM so that one CIM can be translated to many PSMs by defining only one mapping module between a CIM and a PIM. Defining a PIM meta-model can therefore make the automatic translation of the MAIA meta-model platform independent so that MAIA models can be translated to any of the already existing ABMS platforms such as Repast, Netlogo, MASON.



Figure 5.7 – Platform Independent Modelling in MDSD.

We empower MAIA as a modelling tool by building a *second mapping module* that transforms a MAIA model into an AMF (Agent Modelling Framework) model. AMF (AMP, 2013) is a platform independent meta-model that can be used to generate simulations for various ABMS platforms such as Repast, Escape and Ascape (Garro et al., 2012; AMP, 2013)<sup>4</sup>. We will explain the mapping between the MAIA meta-model and AMF meta-model is Section 5.5.

# 5.4 From Model to Simulation

In this section we explain how a MAIA model can be directly transformed (CIM to PSM) into an executable simulation with an MDSD approach.

<sup>&</sup>lt;sup>4</sup>AMP is an interface between modelling languages and simulation platforms. Connecting simulations platforms (e.g., Netlogo, MASON, AgentSpring) and modelling languages (e.g., OperA) to AMP, increases the usability of these tools.

To make a CIM of a socio-technical system with MAIA, we have developed two tools: the web tool that was explained earlier in Chapter 3 of this thesis and a 'maia' plug-in that can be installed on Eclipse<sup>5</sup>. The Eclipse plug-in was developed using the E-core meta-model specification in the EMF framework (Ghorbani and Dignum, 2013) which is one of the most popular languages for developing formal meta-models (Fuentes-Fernández et al., 2012). The E-core meta-model specification is similar to the meta-model presented in Chapter 3 but has an additional package called the *ontological structure*. This structure is the place-holder for building an ontology, so that concepts specific to the domain together with their meaning can be stored and reused as required. Besides this difference, both the web tool and the Eclipse plug-in produce similar XML versions of a MAIA CIM.

We translate the CIM (stored in XML format) to Java code using a set of templates called Java Emitter Templates (JET)(The Eclipse Consortium, 2003). Figure 5.8, shows an example of how executable code can be generated from a MAIA model. The mapping template makes use of the data in the CIM to produce Java code. This code can then be extended to make executable simulation code with additional information regarding initial parameters which we have explained in Section 5.2.



Figure 5.8 – Example of code generation using JET templates and MAIA models.

In this transformation, we use the Java language as our platform specific metamodel (PSM). To develop the mapping templates, the challenging issue is to identify a set of protocols that represent each MAIA concept and relation in Java. One method for identifying these protocols is prototyping (Pavón et al., 2006). Similar to this method, we used our case studies to recursively extract these rules.

<sup>&</sup>lt;sup>5</sup>Eclipse is a platform for developing Java programs (See (Eclipse, 2013)).

### **JET** Templates

In this section, we explain how the simulation specifications described in Section 5.2 are structured into JET templates that produce Java code from a MAIA model. JET templates are a technology to produce Java code from an E-core model in Eclipse EMF. We have implemented 9 main JET templates for this transformation as explained in the following:

- 1. Agent: A MAIA agent transforms into a Java Agent class. All agent classes in the model extend the abstract Agent class. For each agent, we define a boolean for every possible role that the agent can take (is'Role', e.g., isChild). All the physical components (type: private) that the agent owns are also instantiated for the agent. The properties and personal values are all defined as attributes for agent classes. The abstract Agent class has plandId, stepId and willingnessToComply as default attributes.
- 2. Role: Roles extend the abstract Role class. Roles are static classes in Java. They have a method entryCondition() where the agents check whether they can enact a role.
- 3. Objective: For each role, an Objective class is also implemented (e.g., ChildObjective) that will be instantiated for each agent that enacts the role. This class has an objectiveDistance attribute and a method that calculates whether the agent has reached his objective.
- 4. Institution: Similar to roles, institutions are also static. Each institution class has a deonticType attribute and a condition() method.
- 5. Decision Making: A DecisionCriterion class has a condition() method, a MCDAcalculation() method where the actual aspects of a condition and the associated weights are put into a formula, and a result() method. In the result() method, the output of the calculation method is checked against a specified threshold returning a boolean value.
- 6. Physical Component: Depending on the type of physical component, the classes can be instantiated (if type is private) or defined as static where they can be used from any other class in the simulation. If there are composition relations between physical components, the component class that contains the other components will have an instance of the component (e.g., A car has four instances of wheels). A physical component JET template is illustrated in Listing 5.
- 7. Action: Entity actions are represented as action classes in Java, all extending the abstract Action class. Each entity action class has a preCondition() method and an execute() method which is similar to the pseudo-code presented in Listing 4.
- 8. Action Arena: The main class of the Java project calls the ActionArena class. Based on the plan specifications, in the run method of the actionArena class, the agent who enters the method, executes the actions one at a time.

9. Main Simulation: The mainSimulation class has an initialization method, a scheduling method which contains the time loop and agent list, and an evaluation method.

With the JET templates, a Java project is generated in the Eclipse Modelling Environment. The additional requirements to run a simulation is to further complete the main class of the simulation by adding initial values, period of the simulation and number of agents. Therefore, the only part of code that requires further completion is the main class.

```
<%@ jet package="Templates"</pre>
2
  imports="maia.physicalStructure.* maia.ontologicalStructure.*
4 " class="PhysicalComponent_instance" %>
6 package physicalStructure;
8 <%
   maia.physicalStructure.PhysicalComponent pComp =
10 (maia.physicalStructure.PhysicalComponent) argument;
12 boolean allStatic = pComp.getType() == ResourceType.FENCED;
14 String methodPrefix = allStatic ? "static" : "";
  %>
16
  Public class <>>Util.Capitalize(pComp.getName())>>>
18 PhyCom extends PhysicalComponent{
20 <% for (Property prop : pComp.getProperty()) { %>
    <%@ include file="property.inc" %>
24 <% } %>
26 }
```

Listing 5 – A JET template for physical component instances.

# 5.5 Platform Independence for MAIA

In the previous section, we explained a MDSD method that translated a MAIA CIM to a Java PSM. We did this by using a set of transformation rules that we extracted during the implementation of our case studies.

In this section, our goal is to transform a MAIA CIM to a PIM to make MAIA *platform independent* so that modellers would be able to use a variety of simulation platform to produce simulations from their MAIA model.

AMP (Agent Modelling Platform) provides a platform independent meta-model called 'Agent Modelling Framework' (AMF)(AMP, 2013). A model conceptualized

in AMF can so far be automatically transformed to simulations in Repast, Escape and Ascape platforms. AMF is designed to be pluggable and modular so that other developers can create AMF generators for their own tools even if they have no inherent dependencies on Eclipse (e.g., Netlogo, (AMP, 2013)). Figure 5.9 shows the MDSD procedure of simulation generation from MAIA using AMP.



Figure 5.9 – MDSD using AMF

#### 5.5.1 Agent Modelling Framework

The Agent Modeling Framework (AMF) provides high level representations for common ABM constructs. AMF as part of AMP is a platform independent meta-model for agent-based simulation. AMF is analogous to EMF but is targeted toward agent simulations. The foundation of the Agent Modeling Framework is 'Acore' defined in E-core. AMF is fully integrated with the Eclipse platform, but Acore models themselves need have no dependencies on any particular technology beyond XML/XSD (AMP, 2013).

The main concepts of AMF are presented in Figure 5.10. Every model has a *Context. Contexts* are in fact agents that may contain other agents. *Agents* contain *Attributes* such as age and gender. Contexts may contain *projections* which represent some kind of spatial space for the *agents*. Agents are *Actables*. Therefore, they can perform a set of Actions (i.e., *Acts*). At agent level, actions describe behaviours and at context level they can define how projections or agents are created. *Act* has an *execution* setting which schedules the *start* of an execution, its *periodicity* and *priority*.

### 5.5.2 Transforming MAIA Models to AMF Models

To convert a MAIA model into an AMF model, all the concepts and relations of the MAIA meta-model, as illustrated in the class diagram in Chapter 3, should be



Figure 5.10 – The AMF Meta-model - Key Collaborations (AMP, 2013) .

mapped into concepts or relations in AMF. Otherwise, they will not be included in the simulation generated from the AMF model. To do this transformation, we explain which concepts of MAIA fit into each of the main concepts in AMF (Figure 5.10).

#### **AMF** Context

An AMF context is an AMF agent<sup>6</sup>, and contexts can contain other contexts. We use the Context concept to map several MAIA concepts into an AMF model:

- MAIA agent: A MAIA agent is directly mapped into a *Context* (which would also be an AMF *agent*). The properties, personal values and information are defined as Attributes of an AMF agent (and also context).
- MAIA role: An agent can enact a role. Therefore, a role is a context that an agent may contain and the objective of a role is yet another *context* in the role context.
- MAIA role dependency: For every dependency defined between two roles in a MAIA model, a MAIA agent *context* is defined inside another MAIA agent *context* in AMF. In the family care example, an elderly parent depends on his child. In the Java implementation in Section 5.4, we instantiated a family-Member agent (called child) in each agent to define this dependency. Similarly, for AMF we define a child context, inside a family member context.

 $<sup>^6{\</sup>rm Following}$  UML annotations, Context inherits from agent. Therefore all the attribute of the agent become available to the context.

- MAIA physical components: The physical components are also AMF Contexts. The properties of the physical components and the type are *Attributes*. The relation between MAIA agents and physical components is also represented with the concept of *contexts* containing other *contexts*.
- MAIA action arena: The action arena is a collection of *actables* and therefore a collection of *contexts*. Therefore, we define it as a *context* and give the global variables of the system as *attributes* of this context. Some of these global variables can be set to *immutable* attributes if they do not change throughout the simulation. For example, we can define careHomeFee as an immutable attribute of the *Action Arena context*.
- Problem domain variables and validation variables: These are added as *attributes* to the *action arena context*. The type of these attributes is *derived* in AMF because they are calculated from other variables.
- The independent variables defined in the evaluative structure of MAIA are also mapped to *attributes* in related contexts (e.g., agent, action arena, physical components).

#### AMF Act

In AMF, an Act is anything that might happen during the execution of an ABM (AMP, 2013). There are four different types of Acts in AMF illustrated in Figure 5.11: Selection, Command, Builder and Other.

**Commands.** Commands are those acts where an agent who meets the conditions, evaluates some function and based on his state, sets some value. On the other hand, each AMF agent and therefore, each AMF context is an Actable. An AMF Actable contains a set of *Acts*. Therefore, since each MAIA agent has a set of behaviour/capabilities, we define these MAIA concepts as Command Acts in AMF:

- MAIA agents' intrinsic behaviour: These are given to those AMF contexts that represent a MAIA agent.
- MAIA physical components' behaviour: These are given to those AMF contexts that represent a MAIA physical component.
- MAIA roles, institutional responsibilities: These are given to those AMF contexts that represent a MAIA role.
- MAIA agents decision making: These are represented as command acts. The *evaluation procedure* of a command act is overriden to capture agent decision making.
- Calculation of problem domain and validation variables: Since the context is an actable, we define command acts for the calculation of these variables.

Select							
Select	й. С	reate Age	ents∜				
Roots							
Build	≞ R	ule	đ	Wat	ch	ß	
Initialize	₽ <sup>0</sup> S	chedule	ð				
	Query	/		1			
	Q	uery	(7)				
Logic				_			
Logic							
Intersection	@ Ur	nion	- 69	Diffe	rence	600	
Intersection	© Ur	nion	69	Diffe	rence	8	J
Intersection	© Ur	nion	89	Diffe	rence		
Command	© Ur	nion	69	Diffe	etwork	s	
Command State Evaluate	© Ur Sp	ace eave	0	Diffe	etwork Conne	s	
Command State Evaluate	© Ur Sp / L	ace eave Die	00 00	Ne	etwork: Conne Repla	s ect ce	
Command State Evaluate 79 Set 9	y U	nion ace eave Die Move	00 () () () () () () () () () () () () ()	Ne	etwork Conne Replac Discor	s ect ce	
Intersection	Un     Sp     L	nion lace leave Die Move	00 50 60	Ne	etwork Conne Replac	s ect ce	
Command State Evaluate % Set 9 Builders	Un     Sp     L	ace .eave Die Move	00 00 00 00	Diffe Ne	etwork Conne Repla Disco	s ect ce nnect	
Lintersection	Un     Sp     L     L     Bu     Bu     Bu	nion ace .eave Die Move ild Grid ild Geo	。 の の の の の の の の の の の の の	B	etwork Conne Repla Discol uild Sp uild Ne	s ect ce nnect bace 1 et \$	

Figure 5.11 – The four types of AMF Acts and their attributes (AMP, 2013).

**Selection.** In a Selection Act, a rule checks the conditions to see whether agents are allowed to perform Command Acts. We map the following MAIA concepts into a Selection Act:

- Role Entry Condition: For each institutional capability that an agent containing a role wants to perform, the first selection act is to see whether the agent has in fact the right to enact the role and perform the institutional capability.
- Preconditions for entity actions: These preconditions are also defined as selection acts that lead to command acts.
- Institutional Conditions: These are other selection acts that are executed before the decision making command act of an agent.

**Path.** In AMF, the path taken through the program specification is what determines the state of the system. Since the path itself stores the results of prior executions, object (agents) do not need to store values indicating the state of the system after prior executions (AMP, 2013). Following this idea, we define an entity

action as a path where there are various acts. In Figure 5.12, we show the sequence of acts in an entity action which has an institution attached to it.



Figure 5.12 – An Entity Action as a flow in AMF. The second line in each box shows which components of the AMF concept we are using.

Plans in MAIA are also mapped into *flows* (i.e., *paths*) of AMF acts. An atomic plan is represented in the same way as an entity action. A sequence plan is also defined as a path of acts similar to entity actions. To define an alternative plan, we specify a selection act by setting it as a *logic* act with a *union* attribute. A *Union* attribute represents a logical OR joining multiple flows of actions. Finally, a loop plan can be modelled in EMF as a *cause* which is part of a Command act. An action specified as cause occurs upon a specified selection. In other words, this type of action is executed as long as the cause holds.

Similarly, the sanction of an institution is also a flow of acts. It starts from selection acts (entry condition, precondition, institutional condition, sanction condition), and then a command act to execute the sanction. For example, the agent first checks to see if he is the care home owner (entry condition), then he checks to see if he is the end of the month (precondition), then if the fee for an elderly at care home has not been paid (sanction condition), the agent puts the elderly person out of care home.

#### **AMF** projection

All contexts can contain projections. Projections provide visual representations for an agent simulation. The only information MAIA provides for this part of AMF is the connection between physical components and coordinates as properties of these components. However, we can also view the AMF framework as an extension to MAIA models for providing a detailed set of concepts for visualization of simulations.

### 5.5.3 Simulation Details with AMF

The initialization of variables, scheduling and creation of agents as we explained in Section 5.4 are also selection Acts in AMF. These are modelled using the *initialize*,

schedule and create agent attributes of the Selection Act respectively. Furthermore, since selection acts represent sets of agents, the need for a loop to iterate through the agents is omitted (AMP, 2013).

We explained how all the concepts and relations in MAIA can be transformed into AMF concepts. This transformation can be done by hand or automatically. The available software at the AMP website facilitates the automatic translation of AMF models to simulations in Repart, Escape and Ascape platforms.

# 5.6 Conclusion

In this chapter, we described a general architecture for agent-based simulations and proposed methods for developing simulations from high level conceptual models.

We proposed two different model-driven approaches for transforming MAIA models into simulations. First, we defined rules that directly convert MAIA concepts and relations to Java-based simulations. Second, we explained how a MAIA model can be converted to an AMF model, in order to give 'platform independence' functionalities to MAIA.

The procedures and rules we specified in this chapter complement the previous chapters of this thesis as they describe *how* we can make an agent-based model in addition to *what* we can model. This is also a contribution to Agent-oriented Software Engineering (AOSE), because moving from a conceptual model to implementation is not fully addressed in AOSE either (Pavón et al., 2006).

The implementation details we proposed in this chapter are only one way of building simulations from MAIA models. By making different choices, there are many other ways of using MAIA models. For example, the assumption of performing one action per agent per time step can be changed. Likewise, one may decide to have a purely event-driven simulation.

Even though the MAIA meta-model serves its purpose of helping inexperienced modeller and generates complex models with more realistic assumptions, we aim to further expand its functionality by facilitating automatic code generation. To use the methods presented in this chapter for automatic code generation, the specifications of MAIA need to be interpretable by the computer. In the next chapter, we present a formal and detailed version of the MAIA meta-model that would together with the methods presented in this chapter, get us closer to automatic generation of simulations. Moralities, ethics, laws, customs, beliefs, doctrines - these are of trifling import. All that matters is that the miraculous become the norm.

Henry Miller

# Formal Specification of MAIA

# 6.1 Introduction

Simulations provide a dynamic representation of a socio-technical system. A conceptual model on the other hand, describes a static portrait of the system. A conceptual model can only be interpreted by the computer and dynamically represented in a simulation, if it is formally specified. For example, we cannot use a conceptual model of the 'family care system' described in textual format to run a simulation unless every word and its position in the text is defined in a format that can be interpreted by the computer.

Formal specification of a model addresses the concepts, their relations and their positioning in the model in three ways. First, it provides unique and clear definitions for every concept in the model. Second, it provides clear syntax for a model, defining rules which restrict the representation of each concept and the associations between them. For example, the syntactical definition of the English language specifies that there is always the letter 'u' after 'q' in any word spelled in English. Third, it provides the semantics to each concept and relation in that model and their combination.

In MAS research, formally specifying models is common practice. For example, Dignum (2004) provides the formal specification of OperA, Esteva et al. (2001) define formal semantics for E-institutions, and Fuxman et al. (2003) present the semantic and syntax for Tropos. Formal specifications are however, less frequent for simulation models for three reasons. First, as explained in Chapter 5, in most simulation practices, a model is directly described in low level language (i.e., program code) and therefore, there is no need for the computer to interpret high level languages (Pavon et al., 2008). Second, since many modellers make their own ad-hoc models and build simulations from them (Pavon et al., 2008), there is no standard modelling language and therefore, no syntax and semantic to be defined for it. Third, since

the simulation is built by the modeller and there is no automatic code generation, the computer does not need any interpretation process.

In the MAIA simulation platform, we facilitate automatic code generation following model-driven software development (*see* Chapter 5). Since MAIA models are presented in a high level language (i.e., MAIA meta-model), for code generation they need to be interpreted by the computer. Therefore, formal specification of MAIA models is required.

In this chapter, we first explain the formal definitions of the concepts in MAIA that were used to build the syntactical rules (Section 6.2). Since the ADICO structure in MAIA is the most semantically rich aspect, in Section 6.3, we present the semantics of the ADICO institutional grammar. In the same section, we also explain how ADICO can be applied in MAS and finally discuss possibilities for building institutional emergence in agent systems.

# 6.2 MAIA Formal Definition

The syntax of a modelling language<sup>1</sup> is a set of rules that is used to compose a correctly structured model in that language. Since MAIA can be used to build ABMs with and without automatic code transformation, we defined its syntax at two levels: (1) at the level of concept definition in combination with the UML class diagram and (2) at the implementation level together with the e-core model in Eclipse which is explained in more detail in Appendix B. The basis for both syntaxes is the formal definition of MAIA which is presented in this section. As a scenario to explain the concepts, we use the family care example of Chapter 5 here again.

The definition of MAIA is based on three lexicons,  $L_N$  for names,  $L_V$  for variables and  $L_A$  for action bodies. The set of all possible well-formed propositional formulas will be denoted as  $\mathcal{L}_{\mathcal{P}}$ .

MAIA organizes a socio-technical system (STS) into 5 structures:

$$STS = (ClS, CnS, PhS, OpS, EvS)$$

where:

- ClS is the Collective Structure;
- CnS is the Constitutional Structure;
- PhS is the Physical Structure;
- OpS is the Operational Structure; and
- EvS is the Evaluative Structure.

In the following, we will provide a formal definition for each of these structures.

<sup>&</sup>lt;sup>1</sup>The term modelling language and meta-model can be used interchangeably in this context.

#### 6.2.1 The Collective Structure

The collective structure (ClS) describes the characteristics of the community or collective unit of interest:

$$ClS = \{a_1, a_2, ..., a_n\}, for \ n \in \mathbb{N}$$

where  $a_i, 1 \leq i \leq n$  is an agent type.

The agents in the simulation all take an agent type. They are individual or composite entities that make decisions, act and react in a social system. An agent type  $a_i$ , is defined as follows:

 $a_i = (name, Prop_i, PersVal_i, Info, PhC_i, Role_i, IntrCap, DecCrt_i)$ 

where

-  $name \in L_N$  is the name of the agent type;

-  $Prop_i \subseteq Prop$  is a set of properties;

-  $PersVal_i \subseteq PersVal$  is a set of personal values;

-  $Info \subseteq \mathcal{L}_{\mathcal{P}}$  is a set of information;

-  $PhC_i \subseteq PhysicalComp$  is a set of physical assets;

-  $Role_i \subseteq Role$  is the set of possible roles;

-  $IntrCap \subseteq L_A$  is the set of intrinsic capabilities; and

-  $DecCrt_i \subseteq DecCrt$  is the set of decision making criteria.

Agents have properties (e.g., age and gender), personal values (e.g., wealth, health), physical assets (e.g., car, cf. Section 3.4) and information. Agents take roles (e.g., parent, cf. Section 3.3) in the society to perform various actions. They have intrinsic capabilities such as eating and sleeping that are independent of the role they take in the society.

A property,  $prop_i \in Prop$  is a tuple:

$$prop_i = (name, value)$$

where

-  $name \in L_N$  is the name of the property; and

-  $value \in \{boolean, string, number\}$  is the value of the property.

A personal value,  $persVal_i \in PersVal$  is a tuple:

$$persVal_i = (name, value, limit)$$

where

-  $name \in L_N$  is the name of the personal value;

-  $value \in \mathbb{R}$  is the value of the property, and

-  $limit \in \mathbb{R}$  is the limit of value,  $value \leq limit$ .

Agents may also make *decisions* about performing actions. A decision criterion,  $decCrt_i \in DecCrt$  is a tuple:

$$decCrt_i = (name, cond, Aspect_i, threshold)$$

where

-  $name \in L_N$  is the name of the agent type;

-  $cond \in \mathcal{L}_{\mathcal{P}}$  is the condition the agent takes into account for making a decision;

-  $Aspect_i \subseteq Aspect$  is a set of aspects; and

-  $threshold \in \mathbb{N}$  is a value to compare the sum of aspects against.

A decision aspect,  $aspect_i \in Aspect$  is a tuple:

$$aspect_i = (aspectType, weight)$$

where

-  $aspectType_i \in Prop \cup PersVal \cup Inst \cup L_V \cup \mathcal{L}_{\mathcal{P}}$  is the value of the aspect, and -  $weight \in \mathbb{R}$  is the weight that specifies the priority of this decision aspect in the process.

#### 6.2.2 The Constitutional Structure

The constitutional structure is a collection of roles, institutions and their dependencies:

$$CnS = (Role, Inst, Dpn)$$

where:

- *Role* is a set of roles;
- *Inst* is a set of institutions; and
- $Dpn: Role \times Role$  is a set of dependencies between roles.

To be part of a social system (e.g., a family), agents take roles (e.g., role of a father, mother, child, elderly) which places them in certain institutional settings. Each role is created to serve an *objective* in the system (e.g., mother to ensure bringing up a child). If an agent meets the condition to enact a role (e.g., be above 70 years to be in the role of an elderly) certain *responsibilities* become available or acceptable for him to perform. For example, the agent in the role of an *elderly* requests for care or an agent in the role of a *mother* feeds the child. These example also imply that taking roles in a social system creates *dependencies* between roles (e.g., an elderly person depends on her/his child for help). A role,  $role_i \in Role$  is a tuple:

$$role_i = (name, Obj_i, Inst_i, entCon, InsRespn)$$

where

-  $name \in L_N$  is the name of the role;

-  $Obj_i \in Objective$  is a set of objectives;

- $Inst_i \subseteq Inst$  is a set of institutions;
- $entCon \in \mathcal{L}_{\mathcal{P}}$  is the entree condition to enact the role; and
- $InsRespn \subseteq L_A$  is the set of institutional responsibilities.

An objective,  $obj_i \in Obj$  is a tuple:

 $obj_i = (meetingCond, objDist)$ 

where

-  $meetingCond \in \mathcal{L}_{\mathcal{P}}$  is the condition to achieve the objective; and

-  $objDist \in \mathbb{R}$  is the distance left to achieve the objective.

Another important aspect of the constitutional structure, are social rules, norms, or strategies that shape and influence agent behaviour and decision making process. This aspect is called institution (or ADICO institutional statement) as defined in (Crawford and Ostrom, 1995).

An institution,  $inst_i \in Inst$ , is a tuple which describes the principles for acceptable human behaviour in a social setting:

$$inst_i = (Attr_i, deo, aim_i, cond, Sanc_i)$$

where:

-  $Attr_i \subseteq Role$  are the attributes who must comply with this statement;

-  $deo \in \{O, P, F\}$  specifies the deontic: whether the statement is an obligation (O), permission (P) or prohibition (F);

-  $aim_i \in Plan$  is the sequence of actions that is the goal of this institution;

-  $cond \in \mathcal{L}_{\mathcal{P}}$  is the condition; and

-  $Sanc_i \subseteq Inst$  is the set of sanctions for non-compliance called 'or else'.

Table 6.1, shows some examples of ADICO institutional statements in the family care setting.

#### 6.2.3 The Physical Structure

Individuals are influenced by their physical surroundings as well as the constitutional environment. The physical aspects of a social system can be viewed as follows:

$$PhS = (PhC, Cmp, Cnc)$$

where:

- PhC is the set of physical components;
- Cmp is a set of composition relationship betweens physical components; and

- Cnc is a set of connections between the physical components.

A physical component,  $phC_i \in PhC$ , is the building block of the non-social environment that the agents are embedded in:

$$phC_i = (name, Prop_i, type, Beh, Afrd)$$
where:

- $name \in L_N$  is the name of the physical component;
- $Prop_i \subseteq Prop$  is a set of properties;
- $type \in \{private, public\}$  is the type of physical component;
- $Beh \subseteq L_A$  is the set of behaviours; and
- $Afrd \subseteq L_A$  is the set of affordances.

	<b>1</b>	1	1		0
Attribute	Deontic	aIm	Condition	Or	Institution
				else	Type
Child	0	pay for parent	parent no	parent	rule
			money	out	
Child	0	visit parent	every day	-	norm
Child	Р	receive tax return	take care of par-	-	rule
			ent		
Child	-	send its elderly to	parent has	-	shared
		nursing home	alzheimer		strategy
Child	-	hire a nurse	parent is physic-	-	shared
			ally disabled		strategy

Table 6.1 – Examples of ADICO institutional statements in the family care setting.

The elderly parent for example may own a house and the child may have a car or a bicycle. The type of these components is *private* because they belong to a person or a group of people. Physical components can also be shared among everyone in the system (*public*), such as a train, or care home. Each physical component may have *properties* (e.g., colour). Physical components may also have *behaviours* (e.g., ageing) and *affordances* (i.e., what can be done with it; e.g., a bicycle can be ridden).

The other two concepts in the physical structure are composition (Cmp) and connection (Cnc). Physical components may possess a 'has' relationship with other components. For example, the car has tires.

A physical composition  $cmp_i \in Cmp$  is a tuple:

$$cmp_i = (parentCmp_i, ChildCmp_i, cardinality)$$

where:

-  $parentCmp_i \in PhC$  is the physical identifier of the composition;

-  $ChildCmp_i \subseteq PhC$  are the set of physical components that are the elements of the composition, and

- cardinality  $\in \mathbb{N}$  is the cardinality between the elements of the composition.

Furthermore, the connection between the physical components (between road and house, or road and school) may also be required in an agent-based model especially for spatial models.

 $cnc_i = (nodeFrom_i, nodeTo_i, Prop_i)$ 

where:

-  $nodeFrom_i \in PhC$  is a physical component;

-  $nodeTo_i \in PhC$  is a physical component; and

-  $Prop_i \subseteq Prop$  is a set of properties for the connection.

#### 6.2.4 The Operational Structure

The operational environment is viewed as an action arena where different situations take place, in which participants interact as they are influenced by the environment and produce outcomes that in turn affect the environment. The operational structure (OpS) of a social system is described as a tuple:

OpS = (ActionAr, ActionSt, Plan, RoleEnct)

where:

- ActionAr is the set of all entity actions;
- ActionSt is the set of action situations;
- *Plan* is the set of plans; and
- *RoleEnct* is the set of role enactments.

The agents, influenced by the social and physical setting of the system, perform actions in the action arena (actionAr). The action arena contains all the entity actions, entAction<sub>i</sub>  $\in$  ActionAr, that may execute during a simulation ordered by plans which are in turn ordered by action situations (ActionSt<sub>i</sub>  $\subseteq$  ActionAr). Agents take different roles. They perform some actions based on the roles they are enacting at a particular setting (RoleEnct) (e.g., wash elderly person if nurse, request for help if you are elderly).

An entity action,  $entAction_i \in ActionAr$ , is described with the following concepts:

 $entAction_i = (performer, preCon, postCon, postConNotDo, actionBody, Inst_i, RoleEnct_i, Crt_i)$ 

where:

-  $performer \in Agent \cup Role \cup PhC$  is the entity who performs the action: agent, role or physical component;

- $preCon \in \mathcal{L}_{\mathcal{P}}$  is the precondition to perform the action;
- $postCon \in \mathcal{L}_{\mathcal{P}}$  is the postcondition;
- $postNotDo \in \mathcal{L}_{\mathcal{P}}$  is the postcondition if the action is not performed;
- $actionBody \in L_A$  is the name of the activity that the performer executes;
- $Inst_i \subseteq Inst$  is the set of institutions associated to this action;
- $RoleEnct_i \subseteq RoleEnct$  is the set of role enactments to perform this action; and
- $Crt_i \subseteq DecCrt$  is the set of decision making criteria.

A role enactment,  $roleEnct_i \in RoleEnct$ , is described with the following concepts:

$$roleEnct_i = (agent_i, entAction_i, role_i)$$

where:

-  $agent_i \in Agent$  is the agent who performs the action by enacting a role;

- $entAction_i \in ActionAr$  is the action; and
- $role_i \in Role$  is the role the agent is going to enact.

There are four different types of plans that determine the order of action execution. Given plans  $p_1$ ,  $p_2$ , p, entity action a (as an atomic plan), ';' as sequence of plans, ' $\sqcup$ ' as choice and '(*loop*, condition)' as a conditional loop, we define a plan according to the following:

 $plan ::= a \mid p_1; p_2 \mid p_1 \sqcup p_2 \mid loop(p, condition)$ 

- Atomic (a): These plans contain only one entity action (e.g., feed elderly).
- Sequence  $(p_1; p_2)$ : Several actions are executed one after another. After the execution of each entity action, the *step* field of the agent is updated to the next step in this plan *if* it has successfully executed the current step (e.g., wake up, eat breakfast).
- Choice  $(p_1 \sqcup p_2)$ : Choice plans provide an equal probability alternative between a list of actions (e.g., have breakfast or take shower)
- Loop (loop(p, condition)): A loop plan is repeated as long as the condition holds.

These definitions of plans enable a diverse range of action execution. Furthermore, plans are placed in *action situations* to provide the option of defining conditions for a *set of* plans to be triggered.

#### 6.2.5 The Evaluative Structure

The evaluative structure is a collection of validation and problem domain concepts:

$$EvS = (PrD, Val)$$

where:

- PrD is a set of problem domain concepts ; and

- Val is a set of validation concepts.

Problem domain concept,  $prD_i \in PrD$ , is a tuple:

$$prD_i = (dependentVar, independentVar, valueUpdate, relatedAction_i, relationType)$$

where:

- dependent  $Var \in L_V$  is a variable that explains a behaviour in the model;

- independent  $Var \in L_V$  is a set of variables that are related to the dependent variable;

- valueUpdate  $\in \mathcal{L}_{\mathcal{P}}$  is the statement that calculates the dependent variable;

-  $relatedAction_i \in EntityAction$  is the action that influences the value of the dependent variable; and

-  $relationType \in \{direct, indirect, none\}$  defines the type of relation between the dependent variable and the entity action;

The validation concepts,  $val_i \in Val$ , are represented by a similar tuple. See Chapter 5, Section 5.2.1 for examples of these concepts.

In this section we gave a logical representation of the concepts in MAIA as a basis for the syntax. Next, we will provide the semantics for the ADICO structure because of its richness and importance in our simulation platform.

## 6.3 ADICO Semantics

In computer science, semantics is the mathematical study of the meaning of languages by evaluating syntactically correct notations. The semantics field is normally used to build compliers for programming languages.

MAIA builds on the idea of institutions as the backbone of social systems. Therefore, the institution concept in MAIA plays an important role in the type of simulations developed with this modelling language. On the other hand, since we follow the conceptually rich definition of institutions, ADICO by Crawford and Ostrom (1995), it is worthwhile to build the semantics for this part of MAIA.

The ADICO semantics adds two contributions to this line of research. First, as a follow up of this thesis, this semantics is a major step towards *fully automatic* translation of MAIA models into simulations. Second, ADICO as a definition of institutions is new for MAS research. Therefore, since building semantics is common practice in MAS for formally defining agent behaviour and using it in various artificial intelligence application domains (i.e., robotics), by providing the semantics for the ADICO structure, we facilitate the usage of it in MAS research.

In this section, we presents the semantics for the ADICO institutional grammar. Since 'shared strategies' are particularly instrumental in MAS research and have not been defined before, we focus more on this concept, and structure this section accordingly. In the final part of this section, we will explain how ADICO can be used in MAS to reflect more on our contribution to this area of research.

#### 6.3.1 ADICO Statements Revisited

An ADICO statement (cf. Chapter 3) consists of five components namely: <u>A</u>ttributes, <u>D</u>eontic, a<u>I</u>m, <u>C</u>ondition, and sanction (<u>O</u>r else). This decomposition is for the purpose of summarizing and analysing institutional statements<sup>2</sup>, distinguishing between the different types and understanding the formation and evolution of these statements (Ostrom, 2005).

**Attributes** Attributes describe the participants of an action situation to whom the institutional statement applies. Participants can be individuals who are distin-

 $<sup>^{2}</sup>$ We will use 'institutional statement' as a general term to address the concepts norm, rule and shared strategy.

guished by values such as age, sex or even roles in the system. For example, an attribute of an ADICO statement can be a 'parent'. Corporate actors can also be considered as attributes instead of individuals (e.g., care home). These actors can be distinguished by their organizational values such as location and size. The attribute component of an ADICO statement can never be empty. Therefore, if no attribute is specified for a given institutional statement the default value is 'all members' of the system.

**Deontic Type** The purpose of this component is to distinguish between prescriptive and non-prescriptive statements. Some institutional statements do not have any deontic operator. As an example: "The person who places a phone call, calls back when the call gets disconnected" (Ostrom, 2005). For those that do have deontic, the operators are *obligated* (O), *permitted* (P) and *forbidden* (F). While *obliged* and forbidden directly relate to the normative notions of 'ought', 'must' or 'should', per*mitted* seems less related to the intuitive notion of rule. Nonetheless, permission rules, influence the structure of an action situation as weak or strong enforcements. "An act will be said to be permitted in the weak sense if it is not forbidden; and it will be said to be permitted in the strong sense if it is not forbidden but subject to norm." (Wright, 1977, 33). If permission rules are weak, they grant rights to particular participants with certain properties to do an action. This is especially the case for non-compliance to institutions. For example, if the rule is: 'you are forbidden to perform crime otherwise you would go to jail'; one is *permitted* to put a person in jail for not complying with that rule. Since Ostrom (2005) uses the weak definition of permissions, we will also use this definition in this chapter.

**Aim** The aim component describes the action or outcome (i.e., a state of affairs) to which the institutional statement applies. In order for a institutional statement to influence behaviour, individuals must have a choice concerning its Aim. In other words, prescribing an action or outcome only makes sense if its negation is also possible. E.g., the capability of voting implies the capability of not voting.

**Condition** Conditions are the set of parameters that define when and where an ADICO statement applies. If there is no condition stated, it implies that the statement holds at all times.

**Or else** 'Or else' is the consequence of non-compliance to an assigned institutional statement. Only deontic statements include an 'Or else'. A common type of 'or else' is a sanction. Besides sanctions, rule violation may also result in the change of deontic (e.g., forbidden to permitted) of *another* rule. For example, it is forbidden to put a person in jail, but if they perform a crime, then the deontic changes to permission and one is allowed to imprison someone. Institutional actions may also be a result of norm violation. For example the role of the violator may be taken away. In general, the 'or else' component of an ADICO statement contains an institutional statement by itself which results in a nested structure of institutional

statements. Also, the 'or else' component may be linked to the condition component that specifies the number of times that the norm has been violated.

According to Ostrom, an institutional statement can be divided into three different categories namely: rules, norms and shared strategies.

#### 1. ADICO

A  $Rule^3$  (aka, regulatory rule) is the most complete form of statement covering all five components of the ADICO statement. In other words, rules have attributes, deontic type, action, condition and 'or else'.

#### 2. ADIC

A  $Norm^4$  is an institutional statement without an 'or else' component. Although there might be consequences for non-compliance to norms, these consequences are not considered as 'or else' because they are not explicit or unique. For example, shaking hands when being introduced to someone is a norm given that, if not done, it may affect your future relationship with that person. However, there is no fixed sanction and different people may have different reactions.

#### 3. **AIC**

A *Shared strategy* is an institutional statement where there are no sanctions nor deontic type. Shared strategies represent general expectations about the *aggregate behaviour* of others.

#### 6.3.2 Shared Strategies: A Definition for MAS

According to E. Ostrom, a shared strategy is a social concept that refers to a type of behavioural pattern that is observed by a significant number of individuals although it is, prima facie, neither associated with any deontic modality, nor having a reward or punishment linked to its performance. In order to elucidate the distinguishing features of shared strategies, in this section we explore different examples of social behaviour<sup>5</sup>.

Ostrom, in (Ostrom, 2005, pg. 143), proposes as an example of shared strategy, the rule of calling back when a telephone conversation is cut ( $s_1$  in Table 6.2). Strategy  $s_1$  is a conditional that under objective circumstances triggers an action. It does not explicitly entail an obligation or a prohibition, and no explicit or unique reward or punishment ensues. On a closer look, however, strategy  $s_1$  may entail an *expectation*, that, depending on the context in which the interruption took place, may be a strong, possibly asymmetrical and, if not fulfilled may be consequential. The level and nature of expectation therefore reconciles with Ostrom's claim that, if an action rule is to be a shared strategy, then it would not matter whether  $\alpha$  is done or not. We believe that the key is in the collective nature of expectations involved in shared strategies as we shall see.

<sup>&</sup>lt;sup>3</sup>In agent literature, a rule is often addressed as 'norm' or 'regulation'.

<sup>&</sup>lt;sup>4</sup>Sometimes called 'social norm' or even 'moral' or 'ethic code' in MAS literature.

 $<sup>{}^{5}</sup>$ The concept of shared strategies has been addressed by social scientist using different terms (for instance, *scripts* by Schank et al. (1977) or *conventions* by Hodgson and Knudsen (2004)). For an overview of this literature see (Ostrom, 2005, pg. 178).

	· · · · · · · · · · · · · · · · · · ·
$s_1$	When a telephone conversation is cut, call back
$s_2$	When in Rome, do as Romans do
$s_3$	Dutch eat at 5:30
$s_4$	In a busy stairway, walk on the left
$s_5$	Jumping the queue is not nice
$s_6$	Faced with an unexpected obstacle, break
$s_7$	Only when a pedestrian makes a clear sign to attempt to cross
	the street, yield the right-of-way
$s_8$	If no police officer is in sight, skip the red light

Table 6.2 –	- Examples	of Behaviours	that can	$\mathbf{be}$	assumed	shared	strategies.
-------------	------------	---------------	----------	---------------	---------	--------	-------------

Strategies  $s_2$  and  $s_3$  are similar to  $s_1$  but their deontic component is more tenuous and thus closer to Ostrom's intuitive definition. Strategy  $s_2$  "When in Rome, do as Romans do", like  $s_1$ , is an ostensible *directive for action* whose —relatively inconsequential— deontic component may guide the adaptive behaviour of foreigners, on one hand, and the leniency of natives towards non-standard behaviour of foreigners, on the other. Strategy  $s_3$ , "Dutch eat at 5:30", asserts a *factual regularity* but it also hides a directive for action whose compliance by an individual is indifferent to the rest of the world; nevertheless, under certain circumstances, it may have practical consequences (in Holland, for an individual's eating plans or for the operation of restaurants).

These three strategies may be deemed shared strategies only if we make some assumptions about the expectations involved explicit, otherwise they would be examples of *common* and *collective* strategies. Thus, strategy  $s_3$  would not be a shared strategy but a "common strategy" if we understand it as a prevalent behaviour which people may not even be aware of. However, it becomes a "shared strategy" when we understand it as an expectation of common global behaviour; for instance, saying that most people believe that most Dutch eat at 5:30. Finally,  $s_1$  also fails to be a shared strategy when the two parties expect that both parties should follow the rule, or technically, when there is collective belief. That is, we have the three types of strategies characterized in Table 6.3.

	rabie 0.0 briategy rypes
common strategy	most individuals do $s_j$
shared strategy	most individuals believe that most individuals do $s_j$
collective strategy	most individuals believe that most individuals believe most
	individuals do $s_j$

Table 6.3 – Strategy Types

Shared strategies may be situated, thus examples  $s_6$  and  $s_7$  are incompatible if their conditional part is not situated in, say, Holland and Portugal where only one is a shared strategy. Finally, notice that some shared strategies ( $s_7$  and  $s_8$ ) may very well hold and be socially useful in one context but may be highly dangerous patterns of behaviour in others, hence giving rise to full norms that forbid and punish their performance. Situatedness is not only physical as  $s_5$  "jumping the queue is not nice" illustrates. It is a shared strategy in everyday situations like the supermarket or a theatre but becomes a strict directive in surgery waiting lists and in some bureaucratic procedures.

As section 6.3.4 will show, it is important to distinguish between the collective character of a shared strategy –the fact that a collectivity has  $s_j$  or not– and whether each individual decides to enact or not that shared strategy in a particular moment. In fact, asymmetries of different types may create different expectations that affect agents' decisions; for instance, even when  $s_1$  is a shared strategy, if I am calling a cab to go to the airport and communication breaks, it is me who should call back because it is in my best interest to continue the conversation and I may presume the cab doesn't know my number.

Likewise, shared strategies reveal a transient character that puts them between actual standard norms or social conventions, and fully unregulated behaviour, this transient character is revealed both in the collective and the individual perspectives. Thus, from an institutional perspective shared strategies can be seen as an *emerging social convention* or the grounds for an emergent norm. That is the case of  $s_4$ , "walking on the left of a busy stairway", that in London is a solid social convention –whose non-compliance is met with contempt or derision, while in Paris it is a shared strategy, and in the US it is not (still?). Note also that driving on one of either sides of a road, which was a shared strategy at some point, became institutionalized as an explicit norm everywhere; probably because of the social significance of non-compliance. From an individual's perspective, on the other hand, the transient character of shared strategies is evident in the same strategy  $s_4$  that may be likened either to an *internalized norm* or to a *tacit social convention* of which the subject might be not fully aware.

#### 6.3.3 Formalizing ADICO Statements

In this section<sup>6</sup> we formalize the notion of *Institutional Statement* from (Ostrom, 2005) to get to a semantic description of the rules, norms and, foremost, shared strategies of the ADICO definition. To define ADICO statements, Ostrom (2005) follows classical deontic formalizations of Von Wright (1951). Here, we refine this formalization to newest thoughts in deontic logic (Meyer and van der Hoek, 1995; Aldewereld, 2007; Dignum and Kuiper, 1997; Makinson and Van Der Torre, 2000; Dignum, 2004; Broersen et al., 2004).

The logic used for the formalization is a temporal epistemic logic based on CTL<sup>\*</sup> (Emerson, 1990) for the temporal aspect and KD45 (Meyer and van der Hoek, 1995) for the epistemic aspect. We use a technique similar to (Engelfriet, 1996) for the combination of these modalities. In short, the resulting logic is a temporal logic where the states contain an epistemic modality. This allows for the expression of beliefs and changes of beliefs, but not the expression of beliefs about the temporal structures (that is, one can change its beliefs in a future state, but one cannot have beliefs about future or past states).

<sup>&</sup>lt;sup>6</sup>This research is conducted in cooperation with H. Aldewereld, V. Dignum and P. Noriega (Ghorbani, Aldewereld, Dignum and Noriega, 2012, 2013).

The core of the logic is given by the set of propositions  $\mathcal{P}$ , which can be used to construct sentences using the typical propositional operators  $(\neg, \land, \lor, \rightarrow, \leftrightarrow)$ . The set of all possible well-formed propositional formulas will be denoted as  $\mathcal{L}_{\mathcal{P}}$ . This logical core is extended to an epistemic logic of beliefs using a belief-operator (B), following the KD45 principles, resulting in a set of well-formed sentences  $\mathcal{L}_{\mathcal{BP}}$ . The temporal logical language  $\mathcal{L}_{\mathcal{TBP}}$  is then constructed by adding the usual temporal operators: path operators A (all paths), E (some paths), and state operators X(next), G (always), F (sometime), U (until). The language is further enriched with stit:  $e_r$  ('see to it that', see (Belnap and Perloff, 1988)) to express individual action.

To use the logic  $\mathcal{L}_{TBP}$  for the semantics of ADICO, we make some minor modifications to the definition presented in Section 6.2 and reintroduce ADICO statements as follows.

**Definition 1 (Institutional Statement)** ADICO Institutional Statements are of the form

$$D_R(I \mid C) \rightsquigarrow o$$

where

- D represents one of the modalities:  $\{O, P, F, S\}$ 

- R being the attribute, represented as a set of roles;

- I being the aim, represented as an expression from  $\mathcal{L}_P$ ;

- C being the condition, represented as an expression from  $\mathcal{L}_P$ ; and

-  $\rightsquigarrow$  o being the or-else, where o is represented as combination of institutional statements.

The modality of an institutional statement can either be: O (obligation), P (permission), F (prohibition), or S (shared strategy). The modality determines the semantics of the statement. Roles are defined in Section 6.2, with  $\mathcal{R}$  being the set of all roles in the institution. The applicability of an institutional statement is thus  $R \subseteq \mathcal{R}$ . The  $\rightsquigarrow$  o part of the statement expresses the *or-else* of the institutional statement, representing the reaction to violations of the statement. Intuitively, this means that when the lefthand-side of the  $\rightsquigarrow$ -operator is violated, the righthand-side of the  $\rightsquigarrow$ -operator is activated. The reaction, o, is represented as an expression containing institutional statements combined with conjunctions and disjunctions. It is also possible that  $o \equiv \top$ , which expresses that the institutional statement has no reaction<sup>7</sup>.

The different types of institutional statements referred to by Ostrom can be obtained in the following ways. A *rule* is an institutional statement that contains all elements, and where the modality is of deontic nature (that is,  $D \in \{O, P, F\}$ ). *Norms* are institutional statements with a deontic modality ( $D \in \{O, P, F\}$ ) and where no *o* is specified;  $D_R(I | C)$ . Finally, *shared strategies* are institutional statements without a deontic modality (D = S) and where the reaction *o* is absent;  $S_R(I | C)$ .

For the semantics of the institutional statements, we create reductions of the newly introduced operators to the basics of the  $\mathcal{L}_{TBP}$ .

<sup>&</sup>lt;sup>7</sup>Typically, when  $o \equiv \top$ , we omit the  $\rightsquigarrow o$  part of an institutional statement for readability:  $D_R(I \mid C) \rightsquigarrow \top = D_R(I \mid C).$ 

#### **Definition 2 (Obligations)**

$$O_{R}(I \mid C) \rightsquigarrow o \quad \Leftrightarrow \forall r \in R : A \Big[ C \rightarrow (\neg viol(I, r)) \ U \\ (e_{r} \ I \land X(AF \neg viol(I, r)) \lor \\ X(\neg I \land viol(I, r))) \\ \land viol(I, r) \rightarrow o \Big]$$

The above definition transforms the obligation into a  $\mathcal{L}_{\mathcal{TBP}}$  sentence, using an Anderson's reduction (Anderson, 1958), similarly as done in, e.g., (Aldewereld, 2007). Intuitively, the definition expresses that whenever the condition (C) holds, *either* the aim (I) is achieved by those obliged  $(e_R I)$ , in which case no violation of the obligation will ever occur, *or* the aim is not achieved, and a violation happens. Moreover, when the violation happens, the reaction statement *o* (if present) is triggered (these statements typically express sanctioning mechanisms, see (Ostrom, 2005)).

#### **Definition 3 (Reduction of Prohibitions)**

$$F_R(I \mid C) \rightsquigarrow o \iff O_R(\neg I \mid C) \rightsquigarrow o$$

The reduction of prohibitions is based on the principle that  $Fp \equiv O \neg p$  from most deontic logics.

#### Definition 4 (Reduction of Permission)

$$P_R(I \mid C) \quad \Leftrightarrow \forall r \in R : AF \Big[ C \to (I \to \neg viol(I, r)) \Big]$$

The reduction of permissions is based on the principle that  $Pp \equiv \neg O \neg p$  (Boella and Van Der Torre, 2003).

#### Definition 5 (Reduction of Shared Strategy)

$$S_R(I | C) \Leftrightarrow \forall r_1 \in R, \forall r_2 \in R \setminus \{r_1\} : A(C \to B_{r_1}e_{r_2}I)$$

The reduction of shared strategies is formed around the idea that shared strategies represent an *expectation*. Intuitively, a shared strategy expresses the expectation that other members of the same group (i.e., playing the same role, or part of the group of roles that share the strategy) will try to follow the shared strategy. This idea is reflected in Definition 5. This is different from the notions of common strategy, where everyone in the group does the expected thing, and joint strategies, where everyone in the group intends that they do the expected thing. Using similar elements as used in Definition 5, we can also formalize the notions of common strategy and joint strategy:

#### Proposition 1 (Common & Joint Strategies)

$$\begin{array}{lll} CS_R(I \mid C) & \Leftrightarrow & A(C \to \forall r_1 \in R : e_{r_1}I) \\ JS_R(I \mid C) & \Leftrightarrow & A(C \to \forall r_1, r_2 \in R : B_{r_1}B_{r_2}e_{r_1}I) \end{array}$$

Common strategies (CS) happen when all agents in a system are programmed alike, and act in similar manners; that is, every member of a group R follows a common strategy  $CS_R$  to do I when each member of that group does I. A joint strategy (JS), similar to joint-intentions (Dunin-Keplicz and Verbrugge, 2002), is when every member of a group R does I, but also knows (and expects) that every other member of R also does I. That is, there is shared belief that the group believes that they are doing I.

By formalizing the shared strategies (and similarly, common and joint strategies) we lost an aspect of Ostrom's concept. An important aspect of Ostrom's reading is that a shared strategy can be not acted upon, which is missing from Definition 5, since we expect that every agent in the group will do I. Informally, Definition 5 reads as "everyone from group R believes everyone from group R does I". Ostrom's reading of a shared strategy is more in line with "most from group R believe that most of group R do I" (see the discussion earlier in Section 6.3.2). This has an impact on the way agents behave, because in the first reading one can be sure that members of the group R will do I, whereas in the second reading it might be that some members of R will not do I. Therefore, we need to weaken our definition, for which we require a semantic definition of 'most'.

**Definition 6 (Most)** We define the set-theoretic 'most' operator Was follows, for a set of roles R:

$$\mathsf{W}(R) = R' \quad \Leftrightarrow \quad R' \subseteq R \land (|R'| > 1/2 \cdot |R|)$$

Intuitively, this definition expresses what one would expect. If R' is representing the most of set R, then at least half of the agents in R are also in R'; that is, R' is a subset of R and the number of elements of R' is at least half that of R.

Using the concept of 'most' we can create weaker versions of the earlier strategies as follows.

#### Proposition 2 (Weak strategies)

$$\begin{split} CS_R^-(I \mid C) &\Leftrightarrow & A(C \to \forall r_1 \in \mathsf{W}(R) : e_{r_1} I) \\ JS_R^-(I \mid C) &\Leftrightarrow & A(C \to \forall r_1, r_2 \in \mathsf{W}(R) : B_{r_1}B_{r_2}e_{r_1}I) \\ S_R^-(I \mid C) &\Leftrightarrow \\ & A(C \to \forall r_1 \in \mathsf{W}(R), \forall r_2 \in \mathsf{W}(R \backslash \{r_1\}) : B_{r_1}e_{r_2}I) \end{split}$$

The expressions in Proposition 2 represent the weakened versions of the expressions in Proposition 1 and Definition 5. Intuitively, they read as follows. A group R has a *weak common strategy* to I when most of R do I. A group R has a *weak joint strategy* to I when most members of R believe that most other members of R believe that most of them do I. Finally, a group R has a *weak shared strategy* to I when most members of group R believe that most other members of R do I.

A formalization of some of the examples from Table 6.2 is shown in Table 6.4 below.

	Table 0.4 – Examples of Shared Strategies
$s_1$	$S^{-}_{on\_phone}(call\_back \mid conversation\_cut)$
$s_2$	$S_{tourist}^{-}(do\_as\_Roman \mid in\_Rome)$
$s_3$	$S^{Dutch}(eat   5:30)$
$s_4$	$S^{-}_{pedestrian}(stay\_left   in\_busy\_stairway)$
$s_5$	$S^{-}_{civilised\_people}(\neg jump\_queue)$
$s_8$	$S^{-}_{driver}(skip\_red\_light   \neg police\_in\_sight)$

Table 6.4 – Examples of Shared Strategies

#### 6.3.4 Application of ADICO in MAS

In this section, we discuss the practical applications of ADICO in MAS especially shared strategies. Shared strategies can be seen as a form of regulation of individual behaviour within a system, or as mechanisms to improve cooperation, coordination and control in MAS. As such, shared strategies can be used by agents in their reasoning processes, in order to determine their plans in a shared environment (Section 6.3.4), or as means to support design and evaluation of engineered MAS (Section 6.3.4). Finally, ADICO statements can be considered as means to study and build institutional evolution and emergence in agent systems and simulations (Section 6.3.4).

#### **Individual Application**

In this section, we look at how shared strategies can be used by individual agents in their planning. As with norms, agents can and should take into account the shared strategies holding in a domain in order to generate efficient plans for their goals. We assume here autonomous cognitive agents that are able to use their knowledge about a domain in the generation of plans. Such agents can decide on the adherence or not to institutions. Other researchers have also studied norm-based planning (Sofia Panagiotidi and Javier Vázquez-Salceda, 2011) (i.e., the generation of optimal plans with respect to a set of norms). In this section, we concentrate on the use of shared strategies for the generation of plans.

The intuition of the formal definition of shared strategy introduced in Section 6.3.2 is that most agent assumes that under certain conditions, other agents will behave in a certain way. While common strategies may be designed into agent systems so that agents are not aware of them as common behaviour, shared strategies can be perceived by the agents as shared behavioural patterns. If most agents see that most agents have this new perception, the strategies will be globally recognized as shared strategies. This new knowledge will then be updated in their belief system and used in their planning. Based on these new beliefs, agents can take two approaches to use shared strategies in their planning, referred here as an *optimistic* and a *pessimistic* approach. In order to discuss the difference between these two approaches, we take as example the shared strategy:

 $S^{-}_{drivers}(break|obstacle\_in\_road)$ 

which represents the fact that drivers will break when there is an obstacle in the road.

An optimistic pedestrian agent will assume that all drivers will break when he crosses the road, and therefore will plan to cross the road even if he sees a car approaching. On the other hand, a pessimistic pedestrian will assume that you cannot know which drivers will adhere to the shared strategy, since not all have to follow it, and therefore will plan to stop at the curb when he sees a car approaching.

#### Institutional Application

From an institutional perspective there are two issues worth identifying. The relationships between shared strategies and institutional design and evolution, and the role of shared strategies in agent-based simulation.

Since shared strategies constitute a regularity of the aggregate behaviour, institutional conventions may be designed to promote or to control the consequences of that regularity. The approach is straightforward when the existence of a shared strategy is known in advance and it is likely that its execution carries out institutional objectives. In this case, it is reasonable to include specific evaluation mechanisms to monitor the effects of the strategy, and use these to assess transaction costs that would in turn guide the adaptation of the institution to actual performance (Meyer and van der Hoek, 1995). Concomitantly, it is also feasible to establish institutional norms and conventions - with the appropriate evaluation mechanisms - that regiment, constrain or foster the enactment of the shared strategy by participating agents.

The way of dealing with the alternative case is less obvious. When the existence of a shared strategy is not known in advance, ordinary performance monitoring does not necessarily identify the behavioural regularity, even when performance indicators might signal a hidden cost. In such case, institutional reaction may be untimely and ineffectual. To contend with such eventuality, one may attempt to foresee undesirable outcomes and, at the risk of over-regulation, legislate against them. The opacity of undesirable outcomes, however, may sometimes be appropriately addressed with conventional mechanism-design techniques or by a clever use of modelling and simulation methodologies.

In addition to their value for visualizing the effect of shared strategies on institutional performance, in this context, the modeller deals with the system as a regulated MAS, making a shared strategy a feature of individual agents and harnessing individual actions through institutional conventions of different sorts. The use of shared strategies may be fruitful for some forms of agent-based simulation. One relevant form is to use shared strategies as a salient part of the agents' internal decision models. This way, the designer may study different aspects of normative, motivational and goal-directed attitudes (e.g., the interplay of norms and strategies in different agent architectures, norm internalization processes, norm emergence, norm compliance vs. conflict resolution approaches, value formation, achievement degrees). Another form of using shared strategies in agent-based simulation is to factor the analysis of aggregate behaviour by designing populations partitioned by shared strategies, thus measuring cost and value of interactions within populations with pure and mixed strategies, rational or spontaneous triggering of the shared strategies, etc.

#### Institutional Emergence

The ADICO structure can be seen as an instrumental tool to study the emergence of rules, norms and shared strategies in agent societies.

As Ostrom explains in (Ostrom, 2005), the change in any part of the ADICO statement results in the evolution of such entities in a society. For example, when global expectations about a shared strategy narrow down to individuals, a deontic flavour emerges, turning the shared strategy into a norm. Likewise, when the implicit, non-unique and unclear consequences of non-compliance to a norm become common, known and explicit to everyone, that norm turns into a rule with sanction.

To implement institutional emergence, there are three aspects that we need to consider: the implementation of institutions, change of institutions and creation of new ones. To change institutions there are two options. First, the modeller changes an institution (any part of ADICO) and observes how agent behaviour adjusts to the evolved institution. Second, the agents can change the institution themselves under certain conditions. For example, in an agent society, there is a norm that indicates agents must stay on the right side of a staircase. Since this is a norm, there is no sanction for non-compliance but agents may observe different reactions for going to the left. An agent may observe a pattern of behaviour being repeated X number of times. For example, every time an agent stays on the left, he is pushed hardly to the right. The agent who observes this pattern, announces this as a rule with pushing identified as the sanction. If Y number of agents announce the same institution, the new institution (i.e., stay on right *rule*) will replace the old institution (i.e., stay on right *norm*). The evolution of institutions may update any part of the statement (e.g., condition or deontic) and do not necessarily lead to change in the type of institution like the example.

The creation of new institutional statements follows similar procedures. For example, if an agent observes that a pattern of behaviour such as crossing a road at a particular location is being repeated X times, he recognizes this as a shared strategy and announces his recognition. If Y number of agents announce the same recognition, then the pattern of behaviour is established as a new shared strategy in the society.

The discussion above only reflects on the general ideas to implement institutional evolution and emergence. Developing more complex algorithms for the change of ADICO components and agent behaviour regarding institutions can lead to more realistic and sophisticated evolutionary patterns.

#### 6.3.5 Related Concepts to Shared Strategies in MAS

Some concepts in the MAS literature are related to shared strategies. Table 6.5 shows some of the most relevant concepts and compares their usage with similar examples.

Normative information can be *situated* in the environment (e.g., sign boards)

which means that a norm only needs to be followed within a certain boundary of space and time (Okuyama et al., 2007). The type of situated norm can be warning, obligation and direction. A shared strategy however, does not necessarily have to be bound to location and time or have any of the types given to distributed norms (i.e., warning, obligation, direction).

Social conventions are rules that restrict agent behaviour while having no threat or punishment. Peyton (1993) presents the following definition of a conventional norm: "A convention is a pattern of behaviour that is customary, expected, and self-enforcing. Everyone conforms, everyone expects others to conform, and everyone wants to conform given that everyone else conforms."

For a shared strategy however, no one has expectation for others to conform because they are not aware if the person is necessarily a follower of the strategy. No (low) expectation results in no (low) disappointment. For example, if in a given context calling back if the line is dropped is a social convention, then the person may be upset but if it is a shared strategy, the person does not know if the caller is a performer of the shared strategy 'calling back', and thus will not be offended if he does not call back. Therefore, it can also be concluded that a shared strategy has lower priority than a convention for agent planning.

A collective intention is the reason for team existence and it implies that all members intend for *all* others to follow that intention (Dunin-Keplicz and Verbrugge, 2002). The goal of the team may not be reached if one agent decides not to follow the intention. However, for a shared strategy, as mentioned previously, most people know the strategy and know that most others will follow the strategy. Therefore, there is no obligation for agents to perform the strategy and there is also no significant consequence on an individual level while the global behaviour of the system may be important.

Concept name	Reference	Example			
Shared Strategy	(Ostrom, 2005)	The Dutch eat dinner at 5:30			
		pm.			
Situated Norm	(Okuyama et al., 2007)	In this ship dinner is served			
	at 5:30 pm (or else no fo				
Social Norms/ Conven-	(Villatoro et al., 2010)	When eating dinner, people			
tions		start at the same time			
Shared/ Collaborative	(Grosz and Sidner,	Those group of friend have			
plans	1988; Grosz and	plan to make dinner together			
	Kraus, 1996)				
Collective Intention	(Dunin-Keplicz and	Those group of friend are			
	Verbrugge, 2002)	committed to have dinner to-			
		gether at 5:30 pm.			

Table 6.5 – Concepts related to shared strategy in current MAS literature

Norm internalization (Andrighetto et al., 2010) is another topic of research in MAS that can be used in combination with shared strategies. Norm internalization is progressive. This is in line with the transition of ADICO statement from one type

to another (e.g., a norm becomes a shared strategy)(Andrighetto et al., 2010; Ostrom et al., 1994). In other words, during the process of internalization, an ADICO rule which has all five parts of the statement, may lose the 'or else' and become a norm and later on turn into a 'shared strategy' by losing the deontic. On the other hand, the more the norm is internalized the less decision making is required. This again is in line with the definition of shared strategy which is more of a routine that requires less thinking. A fully internalized norm is a shared strategy only if it is shared among people.

The original formulation of shared plans (Grosz and Kraus, 1996) does not see the necessity for an agent to have intentions towards the act of another agent. It is similar to shared strategies in the sense that there is not joint intention between the agents. However, it is different to shared strategies because the agents make plans and actually coordinate in performing the action. Collaborative plans (Grosz and Sidner, 1988) which are a revised version of shared plans are also different from shared strategies because they produce commitment to the joint activity.

## 6.4 Conclusion

In this chapter, we presented the formal specification of MAIA. We explained how we built the syntax for this modelling language, demonstrated a formal definition of concepts, and developed semantics for the most conceptually rich and imperative part of MAIA: ADICO statements.

Formal specifications are important requirements for model-driven software development especially for automatic code generation. Besides targeting this goal, we also demonstrated a contribution to the MAS literature. The ADICO institutional statements provide an instrumental tool for building different types of institutional concepts in artificial systems. They are especially useful because the concepts ranging from culture and norm to strategies and regulations are presented as one common structure. The concept of shared strategy is particularly new to the MAS literature as we explained its application in artificial societies.

As another contribution of this chapter to both MAS and ABMS, we explained the possibilities and preliminary steps in studying and building institutional evolution and emergence in agent systems and simulations. This is an interesting area of research that can be further developed from the material given in this chapter.

# Part IV INSIGHTS

Compare Scripture with Scripture. False doctrines, like false witnesses, agree not among themselves..

William Gurnall

# 7

# An Evaluation Framework for ABMS Platforms

# 7.1 Introduction

As the application of ABMS in social sciences is becoming more apparent, tool development for this approach is evolving into a prominent topic of research. The increasing number of tools however, complicates the choice of modellers in selecting the most appropriate tool according to their requirements.

Since ABMS is a multidisciplinary approach, its tools are also developed with diverse expertise. Therefore, the comparison between tools and the selection of the most appropriate one requires the consideration of various domains. While some of these tools support software development and have a software engineering orientation (e.g., INGENIAS (Pavon et al., 2005)), some support agent development and rely on multi-agent systems (MAS) research domain (e.g., ISLANDER (Esteva et al., 2002)) and others focus on simulation aspects such as data analysis and scheduling mechanisms (e.g., Repast (North et al., 2006)).

Similarly, there are various evaluation methods that compare different features of tools related to specific domains. Some of these methods evaluate software development tools (e.g., (Kitchenham, 1996; Kitchenham et al., 1997; Shehory and Sturm, 2001)). Others are developed for the evaluation of MAS methods and tools (e.g., (Sturm and Shehory, 2004; Cuesta et al., 2003; Dam and Winikoff, 2004; Hesari et al., 2010)). One research, ((Marietto et al., 2003)), specifically provides requirement specifications for agent-based simulation platforms. The main focus of Marietto et al. (2003) is on the practical requirements of a simulation platform such as methods of launching agents or data analysis. To the best of our knowledge, in the current literature, there is no method for comparing ABMS tools that addresses various aspects of such tools related to the different domain requirements (i.e., Software Engineering, MAS and Simulation).

To improve the existing tools, allow the improvement of the next generation of tools, and facilitate the selection of tools based on requirements, we need comparison and ultimately, evaluation methods that are specifically designed for ABMS tools. An evaluation method for ABMS tools addresses three levels of requirement specifications. First, since agent-based models are software systems, the supporting tools should primarily address software development requirements. Second, the social simulations we are addressing are agent-based. Therefore, the tools also need to address the fundamental requirements of agent-oriented software. Finally, simulation requirements such as scheduling and data analysis need to be specifically addressed (cf. (Marietto et al., 2003)).

In this chapter we propose an evaluation framework, addressing the requirement specifications for ABMS platforms<sup>1</sup>. We present the specifications in multiple layers taking on board software engineering and MAS features in addition to simulation requirements. To illustrate the functioning of the framework we compare MAIA<sup>2</sup> with other ABMS platforms.

The remainder of the chapter is structured as follows: in Section 7.2 we address various tools and methods related to ABMS to get an overview of the current state of art in this area. In Section 7.3, we propose our evaluation framework. In Section 7.4, we evaluate a selection of platforms and compare them in Section 7.5. Finally, in Section 7.6, we give the concluding remarks.

# 7.2 Tools and Methods for ABMS

There are various instruments that can facilitate the development of agent-based models. The selection of these instruments, depends on the expertise of the modeller and the domain specific requirements for that simulation. For example, if the simulation is being developed by a computer scientist, the structure of the simulation may receive special attention and a tool that facilitates this would be highly appreciated. However, an inexperienced modeller would prefer a tool that demonstrates results in a seamless manner. Similarly, following the ongoing debate about rich cognitive agents vs. simple agents (Conte et al., 2001; Sloman and Logan, 1999; Axtell, 2001; Dignum, Tranier and Dignum, 2010; Dignum, Dignum, Osinga and Hofstede, 2010), if the purpose of the simulation is to study detailed human behaviour, MAS expertise would provide the possibility to model rich cognitive agents. However, if the goal of the simulation is to get the general behaviour of a crowd, sophisticated human behaviour and interaction may not always be required.

In this section we present the state of art tools originating from different computational disciplines which facilitate ABMS in one way or another. One consensus

 $<sup>^1\</sup>mathrm{We}$  use the term 'platform' to refer to a collection of *tools* and *methods* that together support ABMS.

 $<sup>^2</sup>$  In Chapters 3, 4, we evaluated the MAIA *framework*. In this chapter we evaluate MAIA as a simulation package considering the contents presented in Chapters 5, 6. We do this by comparing it with other simulation platforms found in the literature.

about ABMS in all these disciplines is that *modelling* and *simulation* are recognized as two required practices for this simulation approach (Pavon et al., 2008). However, these two practices are not always distinguished and the process that leads to running simulations is diverse. Therefore, we classify the tools into two groups: those that are geared towards conceptual modelling and the ones that are focusing on simulation implementation.

#### 7.2.1 Conceptual Modelling

Modelling is a prior step to simulation but may even be merged with simulation and take place simultaneously. Since socio-technical systems are complex, tools that support conceptual modelling and address it as an independent procedure are highly instrumental for building simulations (Hassan et al., 2009). Such tools may provide the concepts that need to be put in a model (a modelling language) and offer tools that build the conceptual model.

The INGENIAS (Pavon et al., 2005) platform provides meta-models and software tools to support the conceptualization process of ABMs. The modelling language of INGENIAS is relatively comprehensive, including social concepts such as agent, role, goal, event, society and group. BEMF ((Iba et al., 2004)) is another meta-model for ABMS that is more abstract in the concepts it provides. However, it also addresses the physical aspect of socio-technical systems (i.e., goods) which are not present in INGENIAS. Iba et al. (2004) also provide structured guidelines for the conceptualization process.

The conceptual modelling process in (Garro and Russo, 2010) is also well structured and linked to other functions in the ABMS development process (i.e., design, implementation). The conceptual models in (Garro and Russo, 2010) are the society model, the agent model and the artefact model. The modelling process is also addressed by Grimm et al. (2006) who propose the ODD protocol as a standard way to describe ABMs. Although this protocol is not related to conceptual modelling, it facilitates the reuse and redevelopment of ABMs.

Besides ABMS conceptualization tools, many MAS meta-models can also be instrumental for simulations. The OperA (Dignum, 2004) and MOISE (Hannoun et al., 2000) meta-models are beneficial for those simulations where the organizational structure is an important element. If the distinction between agent and role is necessary in the system, the AGR model (Ferber et al., 2005) and PASSI (Cossentino, 2005) can also be helpful. Finally, where the agents need to follow sophisticated interaction protocols, using Prometheus (Padgham and Winikoff, 2003), OperA (Dignum, 2004), MAS-CommonKADS (Iglesias et al., 1998) and Tropos (Bresciani et al., 2004) is recommended. Prometheus and Tropos also provide clearly specified software development methodologies addressing analysis, design and implementation of the software in distinguished steps. Most of the above mentioned methodologies rely on UML as the language of model specification.

#### 7.2.2 ABMS Platforms

Platforms for *implementing* agent-based simulations are the most common type of tools developed in the ABMS research domain. Netlogo (Tisue, 2004) is one of the frequently used platforms for ABMS. This tool is not limited to social systems and agents (turtles) do not necessarily represent human beings. Therefore, Netlogo is abstract enough to model almost any kind of system. Compared to other ABMS platforms Netlogo is relatively easy to learn and use. Another benefit of Netlogo is the visualization of the simulation and results which make it also suitable for educational purposes.

Repast is another popular simulation platform. It is used within the Eclipse environment and therefore requires Java programming knowledge (North et al., 2006). Repast is normally used for developing more complex ABMs because it offers more flexibility and power compared to Netlogo. While these two platforms rely on agentbased concepts, MASON (Luke et al., 2005) is another java-implemented simulation platform that combines discrete event modelling and ABMS. AgentSpring is also Java implemented, but it does not require programming knowledge for building simulations because it uses graph representations and online information for building simulations (Chmieliauskas et al., 2012). Other simulation platforms are SWARM (Minar, 1996), SeSam (Klugl et al., 2006), Anylogic (Borshchev and Filippov, 2004), Madkit (Gutknecht and Ferber, 2000), AgentScape and Ascape (Parker, 2001). AgentScape is not a development platform but a middleware that supports large scale agent systems and simulations (Wijngaards et al., 2002). None of the mentioned tools (except MASON to some extent<sup>3</sup>) support conceptual modelling. Therefore, an agent-based model is directly implemented as a simulation. This makes the management of more complex simulations difficult because the model is represented in low-level languages. Direct implementation also makes reusability and redevelopment of models more complicated.

One of the simulation platforms that support conceptualization is CORMAS (Common-pool Resources and Multiagent Systems)(Bousquet et al., 1998). While building spatial natural resources is detailed and powerful in this platform, agents are relatively simple entities. MadKit (Gutknecht and Ferber, 2000) is an organization-centred simulation platform that also supports conceptualization limited to the AGR model.

Software platforms that are not especially designed for simulation may still be instrumental. For example, ISLANDER (Esteva et al., 2002) is one of the most complete MAS platforms designed for electronic institutions. It builds on a comprehensive meta-model and supports code generation and thus may be useful for implementing organizations in simulations. Similarly, INGENIAS (Pavon et al., 2008) is not specific to simulations but follows model-driven development and provides plug-ins to facilitate automatic translation of conceptual models to simulation code in Repast. Automatic code generation is also present in simulation platforms such as easyABMS; Garro et al. (2012) translate the conceptual models to AMP, which is then transformed into simulations in Repast, Ascape and Escape platforms.

 $<sup>^{3}</sup>$ MASON has a representation of a simulation (e.g., schedule, agent) rather than a model. We described the differences in Chapter 5.

As we observed in this section, the tools and their functionalities are notably divers. Therefore, to be able to select one, we need a list of requirement specifications to evaluate and compare different tools.

# 7.3 A Framework to Compare and Evaluate ABMS Platforms

In this section we propose a framework to compare ABMS platforms. Since comparing tools is also a method to evaluate those tools, our comparison framework can also be considered as an evaluation framework (Moody, 2005). To develop such framework, we define a list of features that an ideal ABMS platform must support<sup>4</sup>. This framework addresses three sets of specifications: Software Engineering (SE), Multi-agent Systems (MAS) and Social Simulation (SS). In order to identify the most relevant features, we use the state-of-art evaluation research in SE (Kitchenham, 1996; Kitchenham et al., 1997; Shehory and Sturm, 2001) and in MAS (Sturm and Shehory, 2004; Cuesta et al., 2003; Dam and Winikoff, 2004; Hesari et al., 2010; Yu and Cysneiros, 2002; Sudeikat et al., 2005; Cernuzzi and Rossi, 2002; Henderson-Sellers and Giorgini, 2005). We further extend this literature to also cover the SS requirements.

#### 7.3.1 Simulation as Software

A simulation is a software system. Therefore, the platforms that build simulations must have features that support the development of a software system. For the software features layer, we adapt the frameworks proposed in (Dam and Winikoff, 2004; Hesari et al., 2010). These features can be divided into three aspects: those that deal with **modelling language and notation** (Hesari et al., 2010; Dam and Winikoff, 2004; Henderson-Sellers and Giorgini, 2005), those that deal with the **process** of software development (Hesari et al., 2010; Dam and Winikoff, 2004; Henderson-Sellers and Giorgini, 2005) and finally the ones that deal with the **pragmatics** (Dam and Winikoff, 2004) of the platform.

#### Modelling Language and Notation

The core component of any software platform is the meta-model or the modelling language (Dam and Winikoff, 2004). The general features for a modelling language are: *expressiveness*, *preciseness*, *model complexity management* and *user friendliness*.

In terms of expressiveness, the modelling language should address both the dynamic (e.g., data and control flows) and *static* aspects (e.g., structure and knowledge) of the system and be *adequate* in terms of the concepts it is covering (Dam

<sup>&</sup>lt;sup>4</sup>Some of the features overlap with the features discussed in Chapter 3 for usability and usefulness of software tools. Since the goal of Chapter 3 was to evaluate the MAIA framework and the goal of this chapter is to evaluate the whole MAIA platform, we deliberately repeated some of these features as they were both relevant for a framework and a software platform.

and Winikoff, 2004; Hesari et al., 2010). To be precise, a modelling language must have *clear notations* and symbols to represent the concepts with clear *syntax* and *semantics* (Dam and Winikoff, 2004; Hesari et al., 2010; Shehory and Sturm, 2001).

A language should manage the complexity of the model by checking for *consistency* among the concepts (Cuesta et al., 2003), have some form of *modularity* or *hierarchical structure* and *refine* the abstract concepts into more precise and manageable concepts (Dam and Winikoff, 2004; Hesari et al., 2010). Finally, the modelling language should be *easy to use* and *easy to learn* by the modellers. It should also be feasible to *reuse* a model once it has been conceptualized with the modelling language.



Figure 7.1 – Requirements for the modelling language in an agent-based simulation platform.

#### Process

Besides the modelling language, the series of steps that the tool or method takes to develop the software are of great importance. Whether there is a smooth transition between the steps or a semantic gap (Hesari et al., 2010), it is important that the outcome (or product) of each step is clear. One important product of every development step is documentation.

In general, there are seven steps in the life cycle of software development: requirements analysis, architectural design, detailed design, implementation, testing, deployment and maintenance (Dam and Winikoff, 2004). For agent-based simulation, not all these steps need to be addressed in detail. For example, since the graphical user interface of a simulation is not important compared to its content, architectural design and detailed design can be considered as one step that the tool must address. Likewise, simulation software is not a software that necessarily requires maintenance because it is commonly used as a research and understanding tool which would retire after use rather than being kept for long periods of time.

#### Pragmatics

The usability of a software platform greatly depends on the pragmatics of that platform. The pragmatics of a software platform are the most difficult to evaluate (Dam and Winikoff, 2004). Issues related to the pragmatics of a software platform include its quality (in terms of *usefulness* and *effectiveness*) and efficiency (in terms of *time* and *resources*) (Hesari et al., 2010) in building simulations. To ensure the



Figure 7.2 – Requirements for the simulation development process in an agent-based simulation platform.

quality of tools, it is also important to see whether it has been *used by non-creators* and *applied to real* world projects rather than just examples. Also, a simulation requires considerable amount of domain knowledge. Therefore, being able to *involve stakeholders* in the development process is highly instrumental if not necessary.

If the simulation project is large, it is important that the instrument is capable of building larger simulations (i.e., being *scalable*). The tool or method should also be *distributable* among developers to facilitate *team work*. In addition, we also need to consider the managerial aspects of software development (Hesari et al., 2010). These include the consideration of the *risk* of not being able to build the simulation, clear specification of the *roles* involved in the project (e.g., modeller, programmer) and the *scheduling* of the project (i.e., when will each phase of the simulation be deliverable) (Dam and Winikoff, 2004)).



Figure 7.3 – Pragmatic requirements for an agent-based simulation platform.

#### 7.3.2 Simulation with Agents

The concept of agency and fundamentals of MAS research are essential for evaluating agent-based simulation platforms. In this section, we go through the concepts dividing them into three areas: agent attributes, agent communication and agent organization.

#### Agent Attributes

The fundamental concept in the agency theory is *autonomy* (Dam and Winikoff, 2004; Hesari et al., 2010; Shehory and Sturm, 2001; Cuesta et al., 2003). The artificial agents in a simulation should be able to make autonomous decision. Furthermore, agents have *mental attitudes* (e.g., belief, desire, intention, plan) that affect their behaviour and decision making (Dam and Winikoff, 2004; Cuesta et al., 2003). The platform should provide the possibility to build *proactive* and/or *reactive* agents (Dam and Winikoff, 2004; Cuesta et al., 2003; Hesari et al., 2010). Since the goal is to develop simulations of socio-technical systems, the agents may also need to be *situated* in a context which also affects their decision making process (Dam and Winikoff, 2004).



Figure 7.4 – Agent-oriented requirements in an agent-based simulation platform.

#### Agent Organization

The organizational aspects of an ABM are especially important for social simulation because individuals are organized into structures in any socio-technical system. The agents must have *sociability*, meaning that they should be able to interact with other agents (Hesari et al., 2010; Marietto et al., 2003; Dam and Winikoff, 2004; Cuesta et al., 2003). They must take *roles* in the social system and perform *tasks* according to the specified organizational *rules* (Hesari et al., 2010). The agent must also be distributable in the simulation (Shehory and Sturm, 2001; Hesari et al., 2010).

#### Agent Communication

Communication specifications are not a major issue for simulation as compared to agent-oriented distributed software systems. Nonetheless, the general aspects that need to be taken into account are *message handling* and *communication protocols* (Dam and Winikoff, 2004; Hesari et al., 2010; Shehory and Sturm, 2001; Cuesta et al., 2003; Marietto et al., 2003). In addition, dealing with *concurrency* issues may also be crucial for large scale social simulation with multiple threads (Hesari et al., 2010; Shehory and Sturm, 2001).

## 7.3.3 Simulation Features

Some features are specific to simulation and not a major issue for other kinds of software. A major aspect is the implementation of the *environment* in addition to

the agents in the software simulation (Macal and North, 2010; Terna, 1998; Luck et al., 2003; Marietto et al., 2003). The ability to build multiple societies may also be instrumental for comparing distinct settings (Marietto et al., 2003). Since a simulation covers various aspects of a socio-technical system, giving structure to the terminology for that system and developing *ontologies*, can facilitate the reuse of the simulation or at least components of that simulation. Ontologies also facilitate interoperability, knowledge sharing and acquisition (Marietto et al., 2003).

The simulation platforms should support *analysis* of data and provide means of *visualizing* the results (Marietto et al., 2003). It is also helpful if the modeller can observe events during simulation runs (Marietto et al., 2003). Another aspect for a simulation platform is support for *unforeseen (unanticipated)* behaviour of both individuals and the system in general (Marietto et al., 2003).

In the ABMS literature, there are many support tools (e.g., SWARM (Terna, 1998), Netlogo (Tisue, 2004), Repast (North et al., 2006)). This reflects the necessity for a simulation platform to have tools with graphical user interfaces to support inexperienced modellers. These tools would in the least require *scheduling* support and means of *launching agents*. Furthermore, they should also be able to *intervene in behaviour* in an agent simulation (Marietto et al., 2003).

Last but not least, even though we have already addressed validation as part of general software requirements, we stress that it is one of the most important requirements for ABMS platforms. Although a simulation model can never be an accurate representation of a real system, validation ensures that it is at least an appropriate representation according to the objectives of that simulation (Heath et al., 2009). Validation needs to be both *static* (e.g., validation of the concepts through expert interviews) and *dynamic* (i.e., validation of the computational model) (Marietto et al., 2003).

Simulation Features											
Conceptual Analysis						lr	nplement	ation			
Environment	Multiple societies	Ontologies	Data	Event observation	Visualization of results	Detection/ Exploration of unforeseen emergence	Simulation Platform	Scheduling	Launch agents	Intervene in behavior	Static and dynamic validation

Figure 7.5 – Simulation requirements in an agent-based simulation platform.

The features introduced in this section are a list of requirements that an ABMS platform can address. These features can be addressed in different ways, some being better than the others. In the next section we will show how to compare tools given this set of features.

# 7.4 Feature-based Evaluation of ABMS Tools

To illustrate the functionality of our evaluation framework, we have selected three platforms that are most complete in terms of the features they cover: INGENIAS, easyABMS and MAIA.

Using an evaluation form that we prepared based on our feature framework, we asked the developers of each of these tools to evaluate their own platform. We did this in order to minimize bias in the comparison procedure. The evaluation form can be found in Appendix C. The scoring scheme which is adapted from (Kitchenham et al., 1997) is also presented in the same appendix as part of the evaluation form.

In the following, we provide a brief description of each platform and reflect on the outcomes of their evaluation by discussing the benefits and drawbacks of each.

#### 7.4.1 INGENIAS

INGENIAS (Pavon et al., 2005) is a platform originally designed for agent-based software engineering which has been extended to support agent-based simulation. It distinguishes simulation development phases and provides a comprehensive metamodel to conceptualize software in self-defined graphical notations and diagrams. INGENIAS addresses the organizational aspects of a system by defining roles, groups and organizations in different viewpoints. It is one of the few agent-oriented software methodologies that actually addresses implementation and supports code generation (Pavon et al., 2005).

The simulation features of INGENIAS are also quite well-developed. The developers explicitly address different aspects of simulations (e.g., Scheduling) and provide guidelines on how to build a simulation from INGENIAS models (Sansores and Pavón, 2005). It is also possible to translate an INGENIAS model specification to Repast (Pavon et al., 2008).

#### **Evaluation of INGENIAS**

• Software Engineering Features. Since INGENIAS follows model-driven software development, the software engineering aspects of this platform are well developed. INGENIAS addresses the whole development cycle from analysis to deployment. Although the platform itself is well documented, simulations developed with the platform do not come with documentations. INGENIAS covers both static and dynamics aspects of the system under study during conceptualization. The modelling language is in fact quite extensive allowing the modeller to choose the parts that most fit the requirements (Gómez-Sanz et al., 2010). While the INGENME meta-model is used as the syntax of INGENIAS, semantics are less developed.

In terms of model complexity management, INGENIAS performs consistency checks when translating the conceptual model to IAF (INGENIAS Agent Framwork) (García-Magariño et al., 2009). The tool also creates different diagrams to refine the model into modular parts. With the tips and help messages provided, this platform tries to ease the use. It also provides instructions and examples to ease the learning process even though it still remains an issue, especially for those who are unfamiliar with programming.

The simulation project management specifications such as risk management, distribution and project scheduling are not addressed by INGENIAS. However, there are some general role specifications which can be found in (García-Magariño et al., 2009). Even though INGENIAS has been downloaded 13700 times, there is no record of how many people have actually used the platform especially for simulation purposes. Nonetheless, it has been applied in 4 projects so far. INGENIAS does not really involve users and stakeholders.

• MAS features. INGENIAS is primarily a MAS platform, therefore, agentrelated concepts are addressed to a great extent. It is possible to develop reactive agents with mental attitude. Proactivity however, is defined to some extent through internal events incorporating decisions and mental states.

The social aspects of agents are not fully addressed in INGENIAS. There is no support for norms and regulations. INGENIAS defines roles but with a different meaning to organizational roles.

One benefit of INGENIAS in terms of agent-related aspects is support for communication through messages and protocols.

• Simulation features. Finally, the simulation features supported by IN-GENIAS are: modelling of the environment, presentation of simulation data through charts, possibility to observe events and feasibility to represent the code responsible for processing that event. However, INGENIAS does not support detection and exploration of emergence.

In the simulation environment of INGENIAS, agents perform one action per time unit. Agents are launched and managed by the scheduling mechanism. It is possible to intervene in behaviour through the debugging GUI. Finally, while static validation is supported by this platform, validation with stakeholders is not facilitated.

#### 7.4.2 easyABMS

easyABMS (Garro and Russo, 2010) is an agent-based simulation methodology that defines an iterative software development process. easyABMS follows a model-driven approach aiming to support automatic code generation by connecting to the Repast platform. easyABMS uses UML diagrams (e.g., activity diagrams) in the analysis, conceptual and design phases of model development while also proposing its own graphical notations. The developers of easyABMS aim to support modellers with low modelling skills.

#### Evaluation of easyABMS

• Software engineering features.

easyABMS defines the software development process in 6 phases: analysis, conceptual modelling, design, code generation and set-up. This platform has

a modelling language that addresses both dynamic and static aspects of the simulation model by defining separate models such as interaction, behavioural, goal and artifact. All these models build on one reference meta-model which is evaluated to be adequate and expressive even though there are no defined syntax or semantics.

Even though easyABMS has focus on the methodological aspects of simulation development, it relies on the Repast simulation platform for the implementation aspects including execution of simulation, testing, debugging and deployment. This platform does not address the managerial aspects of software development such as project scheduling and does not support teamwork. Documentation is supported to some extent in easyABMS but it is not highest priority.

easyABMS has so far been used by 10 non-developers and applied in two projects. The developers claim that easyABMS is easy to learn, easy to use, easy to reuse and has been specifically designed to support stakeholder involvement.

- MAS features. In terms of agent-orientation, the simulated agents are taskoriented because they are implemented in Repast. However, in the conceptual model agents have goals and it is possible to develop reactive and to some extent proactive agents. Likewise, a conceptual model in easyABMS defines rules and society but does not support distribution of agents. The communication between agents is well captured in the conceptual model. However, in the translation to Repast, all the protocols and messages are limited down to model invocation.
- Simulation features. In terms of simulation, easyABMS models the environment as a collection of artefacts. It is possible to model several societies within one simulation, implementing them as Repast contexts. Since easyABMS uses Repast as the simulation platform, analysis of data is highly facilitated. I.e., it is possible to visualize data, observe events and detect and explore emergence. Furthermore, the scheduling mechanism of Repast which easyABMS uses, is limited to a sort of round-robin non-preemtive type: at a given time, all the behaviours of all agents enabled in that time step are 'ordered' and executed one by one.

# 7.4.3 MAIA

MAIA is a simulation platform that follows a model-driven development approach to build simulations from high-level model descriptions. The main goal of this platform is to provide a comprehensive modelling languages that covers a diversity of concepts in socio-technical systems. In addition to the modelling language (i.e., meta-model), MAIA also provides tools to support the conceptualization process and transformation protocols to covert a conceptual model into a simulation. A supporting software also facilitates semi-automatic translation of models to simulations.

#### **Evaluation of MAIA**

- Software engineering features. Since MAIA follows model-driven development, it supports the software requirements layer to a great extent. One of its major benefits in this layer is the modularity and refinement of the concepts which make model complexity more manageable. However, the modelling language does not have full semantics and therefore with the current level of specifications full automatic code transformation is not feasible. In terms of the process of software development, MAIA addresses the different phases of software development with a smooth transition which means that the phases are not clearly separated. The tables and diagrams in the MAIA interface provide a useful means of documentation. In terms of pragmatics, MAIA is a useful asset for teamwork. It facilitates distribution of tasks and provides an interface between members of the development team. However, MAIA does not address risk management or project scheduling in anyway.
- MAS features. It is possible to develop reactive and proactive agents in MAIA with sociability. Furthermore, organizational aspects such as roles and rules are fully addressed in MAIA. However, there is no explicit representation or facilitation for agent communication: MAIA does not address communication protocols in any way nor does it deal with concurrency problems.
- Simulation Features. Finally, the simulation requirements that MAIA fully addresses are the simulation of the environment, the storage of all data produced in the simulation, the scheduling, the static validation (communication with domain experts) and dynamic validation (evaluative structure). Intervening in behaviour while a MAIA simulation is running is not yet build in and there is not automated method for detecting unforeseen emergence.

All three platforms address the three layers of the evaluation framework but each one is stronger in some aspect than the other. A comparison between these three platforms is discussed next.

# 7.5 Comparison between the Four Tools

In the previous section we discussed the most important features of three simulations platforms and reflected on the main limitations of each. In this section, we will compare them to show the suitability of each platform for different situations.

**Software Engineering Features** All the three platforms are advanced in terms of software development phases even though they are less developed in pragmatics such as risk management and project scheduling.

The main benefits of MAIA for building simulations compared to the other two platforms are: (1) it has proved to facilitate stakeholder involvement, (2) it has been applied to higher number of real projects and (3) it has been used by higher number of non-developers. The main benefits of INGENIAS compared to others is that implementation and deployment especially automatic generation of code is well developed. The main benefits of easyABMS in this respect is that it connects to Repast and relies on many of its features for simulation development.

Therefore, it appears, that in situations where the modeller does not want to do any programming, INGENIAS may be a better solution. Nonetheless, the modeller would still require to learn INGENIAS as it is a relatively complex platform. If the modeller knows the Repast platform, the most efficient solution is to use easyABMS for making the conceptual model. MAIA, is especially suitable for large real world projects, where stakeholder involvement is essential. It is also a better choice if there is a programmer in the simulation development team since there will be a higher chance that the simulation is built in a professional manner both conceptually and technically.

**MAS Features** The main benefits of MAIA in terms of MAS features is that it covers many more agent-related concepts including role, institution and task. In MAIA, it is also possible to model proactive and reactive agents with mental attitude. The main benefit of INGENIAS is the specification of the communication protocols and messages. Finally, easyABMS is better in terms of concurrency management because it relies on Repast.

In situations where the modeller is an expert in MAS, using INGENIAS is the best option because she would be able to use the communication features of this platform. If the simulation is modelled across machines, easyABMS can facilitate concurrency management. Finally, in projects where modelling complex behaviour of individuals is required, the social aspects of a system are being analysed or the goal is to compare policies, MAIA would be beneficial.

**Simulation Features** The benefits of MAIA over the other two platforms are: (1) modelling of the environment, (2) possibility to build multiple societies,(3) support for ontology and (4) possibility to statically validate the model by communicating with stakeholders. INGENIAS is better in (1) tool support, (2) ability to intervene in behaviour and (3) to some extent, data handling. easyABMS is a better tool for (1) data handling as it relies on Repast, (2) it models the environment and (3) provides dynamic validation, again relying on Repast.

With the current developments for each of the three platforms, easyABMS enables a more complete data analysis module which is essential for every simulation.

Although there are many platforms, tools and methodologies for agent-based simulation, we compared three of the most complete platforms in terms of feature coverage. This does not imply that they are the best simulation tools available because there are other simulation tools that are relatively strong in one aspect while not covering other aspects at all. For example, Repast is one of the most powerful tools in terms of implementation and data analysis while it misses many features such as software development steps and a modelling language. Therefore, we would like to stress that the purpose of the evaluation framework is to allow technological comparison between ABMS platforms in order to select the most appropriate tool for the purpose.

# 7.6 Conclusion

In this chapter we presented a framework to evaluate and compare ABMS methods and tools. This framework was presented in three layers to cover software development, agent-oriented and simulation features.

Given the availability of many simulation platforms, our evaluation framework facilitates a technological comparison between ABMS platforms, allowing modellers to select the tool most fitting their requirements. Furthermore, our framework allows for the evaluation of ABMS instruments in order to identify the drawbacks and limitation of each for further improvement.

By using the evaluation framework and comparing MAIA with two other simulation platforms namely: INGENIAS and easyABMS, we detected the major areas where MAIA could be further developed and extended. These include data analysis, more advanced support tool and project management details. We will discuss the details of these limitations in the conclusion chapter of this thesis. All our knowledge begins with the senses, proceeds then to the understanding, and ends with reason. There is nothing higher than reason.

Johann Wolfgang von Goethe

# **B** Discussion and Conclusion

#### 8.1 Overview

Socio-technical systems are a collection of heterogeneous decision making entities and technological artefacts. These systems are governed by public policy within an institutional context, ranging from norms and values to technical standards. ABMS is a suitable exploratory approach to study socio-technical systems in order to gain insights and investigate the possible outcomes of policy interventions. However, to understand various levels of behaviour in socio-technical systems and increase the usability of ABMS, we needed to overcome conceptual and practical limitations of this approach. This motivated us to formulate our research question as follows:

How can we build social structures in agent-based models and increase the utility of ABMS for policy analysis?

In this research we showed that a combination of social theories and frameworks can be structured and formalized in such a way as to incorporate social structure into agent-based models. At the same time, this practice also increased the utility of ABMS to support decision making for socio-technical systems which we demonstrated through case studies.

In this research, we designed a modelling framework to give structure to agentbased models of socio-technical systems and to specifically incorporate social content into these models. We provided tool support for the modelling framework in order to increase the usability of ABMS for modellers with different levels of expertise especially those with less experience in programming. Finally, the modelling framework and tool were designed to facilitate participatory model development.

The overall outcome of this research is the MAIA modelling platform which con-
sists of (1) a modelling language (i.e., framework) for conceptualizing agent-based models of socio-technical systems, (2) a web tool that facilitates the conceptualization process especially participatory model development, (3) transformation protocols that enable the translation of MAIA models into simulations and (4) software modules that facilitate semi-automatic translation of MAIA models to simulations.

In Section 8.2, we will briefly summarize the development process of MAIA. In Section 8.3, we will discuss the outcomes of this research. In Section 8.4, we will reflect on the major findings. In Section 8.5, we will discuss the lessons learnt. Finally, in Section 8.6, we will suggest areas to further improve and extend this research.

## 8.2 Development Process of MAIA

**Rationale.** To incorporate social structures into agent-based models, we started out by developing a conceptual framework. As we explained in Chapters 1 and 2, institutions are one way of viewing social structure in socio-technical systems. Therefore, we adapted the IAD framework as the basis of this research in order to model institutions in ABM and specify the link between this concept and other components of a socio-technical systems. Since some concepts in IAD were not clearly defined or formulated, we used three other social theories and frameworks, namely: structuration theory, social mechanism and actor-centred institutionalism to add further details to IAD.

To build the ABMS framework, we described a typical socio-technical system with the IAD framework and used explanations from other theories, where IAD was too abstract or vague. The result of the integration of these analytical tools is the MAIA framework (*See* Chapter 3).

**Functionality.** We relied on software engineering principles and more specifically model-driven software development to link MAIA models to simulations. First, we proposed a simulation architecture that MAIA models would transform into, and demonstrated mechanisms that transform MAIA models into simulations (*See* Chapter 5). To increase the usability of MAIA, we provided additional transformational mechanisms that link MAIA to a platform independent framework called AMF. This would for example, facilitate building Repast simulations from MAIA models. Furthermore, we built a software module that generates code which can be further completed to make running simulations. Second, MAIA was formulated as a software meta-model. We provided formal definitions, syntax and partial semantics that together facilitate generation of simulations whether by a programmer, or a translator software (*See* Chapter 6).

**Usability.** During the development process of MAIA, we took the intended users of MAIA who are social scientists and policy analysts, into account (*See* Chapter 3). In general, these users have less experience in programming and computational science. Therefore, while keeping the level of modelling language high for the users

to understand, the concepts had to be rigorous and formally defined for simulations. Besides the concepts, MAIA also consists of numerous relationships between these concepts. Keeping track of these relationships during model development was strenuous. Therefore, to ease the use of the framework and manage relational complexity, we developed a user interface (i.e., the web tool) that supports the users by automatically generating relationships and checking for consistency among concepts. We also prepared documentations and guidelines for making conceptual models and for building simulations both at a generic level and a detailed level (cf. (Railsback et al., 2006)).

Having provided support for inexperienced modellers, our final objective was to facilitate participatory model development which we achieved in three ways:

- By using a modelling framework that is described in a high level modelling language, domains experts and problem owners are able to collaborate in model development (cf. (Ramanath and Gilbert, 2004; Railsback et al., 2006)).
- The user interface of MAIA is designed online without having any installation requirements so that stakeholders can freely use the tool. In addition, by saving MAIA models via google drive, the modellers are able to share models and build them collectively.
- The MAIA framework is presented in structures to provide a modular architecture for modelling socio-technical system which also facilitates collaboration (cf. (Ramanath and Gilbert, 2004)) both on domain expert and programmer sides.

**Evaluation.** In the final stage, we evaluated MAIA in three steps. First, the MAIA framework was evaluated as a conceptual framework. We evaluated its completeness by comparing it to IAD. We also looked at it in terms of soundness and parsimony. Since participatory model development and user experience were our main objectives, we looked at the usability and usefulness of MAIA more in depth. This was done by conducting case studies and incorporating user feedback through questionnaires. In the final step, we evaluated the whole package of MAIA as an ABMS platform by proposing a framework that evaluates and compares such platforms. The outcomes of this research process are explained in the next section.

## 8.3 Research Outcomes

We have answered the research question by achieving the three objectives of this research:

- 1. To develop a conceptual framework for describing a socio-technical system and formalize it for building computer simulations.
- 2. To build a tool and provide simulation development guidelines for social scientists and policy makers with different levels of expertise in programming and simulation.

3. To enable participatory ABMS from the early conceptualization phase of the simulation process.

Each of these objectives will be discussed in the next three subsections.

## 8.3.1 The MAIA Modelling Framework

The first outcome of this research is the MAIA modelling framework. MAIA conceptually describes socio-technical systems and incorporates social structures into ABMs. Therefore, these structures as part of socio-technical systems can influence various components of the system, and be influenced by them.

The case studies and many examples we went through, provide evidence that MAIA is comprehensive enough to model socio-technical systems from a variety of domains (*See* Chapter 4). By comparing this framework with other modelling frameworks, we came to the conclusion that MAIA covers more concepts of a socio-technical system than any other existing framework that we have found in the liter-ature. This is mainly because MAIA builds on the IAD framework which has been in development for many years, identifying concepts that are abstract enough to act as a high level ontology for socio-technical systems. Future case studies may yet reveal lack of comprehensiveness in MAIA. Nonetheless, with the systems we have explored so far, MAIA does explain a diversity of socio-technical systems.

Besides a comprehensive modelling framework that helps decompose, conceptualize and analyse socio-technical systems, the ability to build simulations from a high level modelling language is in itself another contribution of this research. Although there are many high level modelling languages especially in MAS domain, being able to produce software is not practised for all. As part of the MAIA platform, we provided guidelines, protocols and software modules that translate a model described in a high level language to an executable software (*See* Chapter 5). Finally, MAIA can also be considered as a documentation tool that standardizes descriptions of agent-based models. This facilitates reuse and redevelopment of such models which is a major requirement in the current state of art.

## 8.3.2 Supporting Users with Different Levels of Programming Expertise

The MAIA platform supports modellers with various levels of programming expertise. The MAIA meta-model is described in a high level language using concepts that are frequently used for policy analysis. Therefore, making a conceptual ABM does not need any programming knowledge.

In general all users of this platform benefit from a predefined template which gives them a set of concepts that may be relevant to model a socio-technical system. A conceptualized MAIA model can even be sufficient to analyse the system and draw conclusions before building a simulation which happened in one of our later case studies<sup>1</sup>.

While conducting case studies, six people who were less familiar with simulation and programming successfully used MAIA and indicated that without it, they would not have been able to make models. The conceptual models these modellers made were sufficient to build computer simulations without acquiring additional domain knowledge. Those users who had experience in programming have also found MAIA useful because it guided them through the concepts that may be relevant to consider in a simulated system. Furthermore, the transformation guidelines and implementation protocols showed them how to make a simulation from a MAIA model. Using MAIA however, does not mean that the user can arrive at running simulations straight away. There are two options : (1) the modeller can work in a team where there is a programmer who would take the MAIA model and make a simulation, and (2) the modeller can get automatically generated code, which would need additional details for a running simulation. Even with the second option, programming is substantially reduced to minimum input.

## 8.3.3 Participatory ABMS

To use ABMS for policy analysis, social scientists, policy analysts, problem owners and domain experts can collaborate to gain a better understanding of the system and, compare and evaluate various policy implementations. These stakeholders can all be involved in the simulation process or at least be able to understand and follow what is being put in the model and what comes out of it.

An implication of MAIA as a predefined structure for modelling, is that it abstracts away from simulation details to organize an agent-based model in a language that is understandable to different stakeholders, motivating them to participate in the development process, trust the simulation and use its outputs. As the case studies showed (*See* Chapter 4), it is possible to involve a variety of people in the model development process besides the modeller of the simulation. Not only this involvement buys the trust of stakeholders, it also helps build more reliable models because the concepts that will eventually result in a running simulation have been verified with the problem owners and domain experts.

Besides involving stakeholders, participation is also highly instrumental in managing larger and more complex simulations because it facilitates the separation of tasks as we saw in our case studies. By using MAIA, people with different expertise can contribute to model development and communicate in a standard language. In general, MAIA structures information to facilitate communication during model development: (1) the modeller can communicate with problem owners or other analysts to explicate the information she will be putting in the model and verify it, (2) different modellers can cooperate in building the conceptual model since the system is well structured, (3) the modeller can give a MAIA model as the outcome of the conceptualization process to a programmer to build a simulation. Therefore, the

<sup>&</sup>lt;sup>1</sup>Investment in greenhouses was another ABMS project where MAIA was used to build a conceptual model (Schrauwen, 2012). The modeller used Ethnography to make the conceptual model. Structuring and decomposing the system into a MAIA model helped her gain a structured view of the system and analyse it to draw conclusions without building a simulation.

programmer does not need to know any domain knowledge and the modeller does not need to know programming. In the simplest form, a domain expert can make a conceptual MAIA model, a programmer can build the simulation and an analyst can analyse the results of the simulation.

Separating modelling and programming is not straight forward. In many cases, either the conceptual model is too formal and computationally complex, requiring computer knowledge, or vice versa; the programmer would need to gain domain knowledge in order to be able to make the simulation because the concepts are usually too abstract, vague and informal. With MAIA, people with different backgrounds can focus on their own area of expertise which results in: (1) more professional models, (2) efficient use of time and resources and (3) more reliable results due to less error proneness.

Considering the MAIA platform as the general outcome of this research, we claim that MAIA simulations can be more insightful and trustworthy than other types of simulations because of the following reasons:

- All the assumptions that structure a MAIA simulation are taken from theoretical research that explain the structure and content of socio-technical systems instead of ad-hoc choices which are common in ABMS.
- MAIA simulations can be conceptually verified with problem owners because of their explicit structure and thorough documentation.
- Since MAIA models are described outside simulation code, the simulations can be replicated more easily, increasing their reliability.

In conclusion, this research is a first step in the process of bringing ABMS within the reach of social scientists and policy analysts and putting this simulation approach to its full potential. Even though we were able to make simulations for all of our case studies, we learnt that there are situations where MAIA is more applicable or useful than other tools:

- MAIA is specifically designed for *social* systems. Therefore, other types of complex systems such as biological systems cannot not be modelled with MAIA.
- MAIA is useful where modelling institutions and social structures is an important element for the problem under study. This particularly holds for policy problems because policy instruments are designed as purposive design of institutions. If the goal is to study a simple generic behaviour or model an abstract system, building a MAIA model may have too much overhead.
- MAIA is meant to facilitate modelling complex socio-technical systems. If the goal is to study an over simplified system that can be easily implemented in a simulation platform such as Netlogo, it may not be necessary to use MAIA or even conceptualize the simulation model to this extent.
- MAIA is geared towards *exploration* of the effect of policy instruments on individual and system behaviour. Therefore, like any other ABMS tool, it

does not provide prediction of outcomes but only gives insights into possible futures.

Finally, 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, judgement, creativity and originality of the modeller.

## 8.4 Reflection

In this section, we reflect on the process and outcomes of this research.

## 8.4.1 Multi-disciplinary Approach to Understand Complex Socio-technical Systems

In this research we were able to use a combination of three distinct fields of research namely: social science, software engineering and artificial intelligence. More specifically, we applied theories and frameworks used in policy analysis in combination with MAS research in artificial intelligence and MDSD in software engineering to develop the MAIA platform. We believe that this research would have been impossible if any of these pillars had not been present.

In order to conceptualize agent-based simulations of socio-technical systems and incorporate social structures into simulations, we required reliable assumptions which we took from the result of years of research in the social sciences. We relied on this scientific domain especially institutional economics to make the general assumption that all socio-technical systems have an institutional backbone. Following that, the structure we gave to a socio-technical system and the components and relations we defined, are all relying on assumptions that are one by one taken from theories and frameworks in the social sciences.

Explaining a socio-technical system with a set of theories and framework is in itself a contribution to analytical studies of such systems but building simulations required knowledge from the artificial intelligence domain where artificial societies have been developed for many years. Combining institutional research in the social sciences with multi-agent systems research in artificial intelligence not only contributed to building rich simulations, it also contributed to the MAS domain by adding more content and structure to artificial societies especially in terms of institutions (*see* (Ghorbani, Aldewereld, Dignum and Noriega, 2013)).

Finally, linking theoretical descriptions and assumptions to simulations and keeping the level of abstraction intact, was only possible by using model-driven development techniques from the software engineering domain.

## 8.4.2 Future of ABMS

ABMS is a simulation approach that has been purely *bottom-up*, observing emergent outcomes as a result of interaction between *simple* entities called agents (cf. (Epstein,

2006)).

The level of complexity in agents (simple vs. rich cognitive) has long been a debate among scientists. The general agreement on this issue is that the complexity of the agents highly relies on the purpose of the model. While some simulations require extremely simple agents (e.g., traffic simulations, crowd behaviour), other simulations require more sophisticated intelligent entities who make complicated decisions (e.g., bio-gas investment). However, besides the level of complexity within agents, the content of the agent simulation and the structure in which agents interact to produce emergent outcomes is another issue worth considering.

As we saw in this research and as we have seen in major debates between scientists (e.g., (Conte et al., 2001)), independent of the purpose of the simulation, agents require structures within which they perform actions. In current ABMS research, even if there are social rules that need to be considered in the simulation, they are modelled as part of the agents. However, social structures exist in every human system, independent of individuals. In other words, an agent-based model that merely consists of a set of agents cannot reflect a real world socio-technical system because all systems in reality contain social and physical structures that influence agents and are influenced by them. Therefore, agent-based models require primary structures that give context to agents to interact in.

One contribution of MAIA as an ABMS platform is that it views agent-based models as a collection of agents in social and physical structures, thus providing a more natural environment for agents to interact in. This system architecture is supported by social scientists (e.g., the bathtub model (Coleman, 1986)) and artificial intelligence scholars (e.g., (Conte et al., 2001)).

## 8.4.3 Institutional Emergence

Emergence is a prominent topic of research both in ABMS and MAS. In fact, in all definitions of ABMS, emergent patterns or outcomes are addressed as the consequence of agent interaction, and the expected result of an agent-based model.

Another contribution of MAIA is that it facilitates institutional emergence. By integrating institutions in a structured and explicit manner in agent-based models, it is possible to model and observe evolution and change in institutions. Therefore, it becomes more feasible to detect, analyse and possibly control institutional emergence (*see* Chapter 6, Section 6.3.4).

Being able to create, detect and observe emergence is particularly important in agent-based simulations because the whole purpose of making agent-based models is to see how patterns and structures emerge as the result of agent interaction. Since our simulations are for the purpose of policy analysis, institutional emergence is particularly important to explore the long term consequences of policy implementations.

We did not explore institutional emergence in any of our case studies because it was out of scope, considering the goal of this research, but this certainly is an area to further extend these case studies and explore new possibilities in the simulations.

## 8.4.4 A test environment for MAS

Besides giving insights into socio-technical systems to address policy problems, social simulation also serves a purpose in MAS research. After building an agent-based system (e.g., intelligent traffic light), the software needs to be tested in the environment (e.g., a city). It is costly and sometimes infeasible to test the software in a real environment. Therefore, the software developer builds artificial environments to test the agent-based system. MAIA can be used to model socio-technical systems as virtual environments to test agent-oriented systems.

## 8.5 Lessons Learnt

The goal of this research was to make conceptually comprehensive agent-based models by incorporating social structures. In addition, we aimed to broaden the accessibility of ABMS and facilitate participatory model development.

We approached this goal by using institutions as the building blocks of social structure. There were other options we could have used such as psychological theories or cultural studies. Since psychological studies have more focus on individual behaviour and have frequently been applied in MAS, for the purpose of modelling social structures, we decides that institutions would be more relevant. Studies on culture would however be highly beneficial for the type of problem we are addressing. Nonetheless, they would mostly provide insights into the informal structures of a social systems rather than formal aspects such regulations and laws. Since we are aiming to support policy making, formal structures also required special attention and thus institutions seemed more relevant.

Having selected institutions, we used IAD to explain socio-technical systems, not only to describe institutions, but also to explain their link with other aspects of the system. There were various options that we could have taken in this path. As one option, we could have used ADICO (part of IAD) to explain institutions and explain the other aspects of a system from scratch, using different theories and frameworks where relevant. While this option would have given us more flexibility in defining the different aspects of a socio-technical system, it would have been less reliable and more difficult to validate. IAD is the result of many years of experience and has been evaluated multiple times. Getting to the level of richness in IAD would have therefore taken considerable time. One other option would have been to use a totally different institutional framework such as Williamson's four layers of institutions or actor centred institutionalism. The main advantage of IAD over these frameworks is the clear structure for institutions, which proved to be highly instrumental for our research.

To support our conceptual framework, we developed an online web application which is relatively simple and easy to use. In order to be able to use the software in a web browser and not require additional installation, we had to sacrifice extra functionalities such as diagram options or data storage format. Although a standalone software would have been more powerful, it would have limited the accessibility for the users because it would have required the installation of software such as JVM. The users we are aiming at are less experienced in programming and software. Therefore, a simple environment that can be accessed on a web browser is more appealing to users who are merely looking for a way to perform analytical studies rather than building professional software. Nonetheless, to also facilitate professional software development with MAIA, we developed the Eclipse plug-in for the more experienced users which is still not a standalone software but has more functionalities.

One final choice that we made was regarding the flexibility in making MAIA models especially with the web application. The input to a MAIA model is quite open and flexible. It is almost like putting free text in the model. This could have been restricted so that the modeller would only be able to specify predefined descriptions in the model. This would have enabled more complete translation of code which is in fact very instrumental in this area of this research. However, it implied that the user would need to learn a modelling language which is at a higher level than programming languages such as Java or C but would still require extra learning effort; something that not all modellers would appreciate. The way we have formalized MAIA is just enough to build a clear and unambiguous model so that a programmer would be able to make a simulation, but not detailed enough to enable fully automatic translation of code. The main benefit of this is that learning efforts are exceptionally decreased so that MAIA is more appealing and less tedious for people to actually use.

## 8.6 Future Work

In this final section, we will propose several areas for future work.

Advancing tool support. Although the web tool has proved to be useful in supporting MAIA, its improvement can further ease the use and learning procedure of this framework. The MAIA web-tool can be redesigned to make it more user friendly, for example, by providing online tips when putting information in. Currently, automatic generation of code is enabled by restricting the users in the information they fill in, within the Eclipse environment using the MAIA plug-in. Restricting the web tool in a second version, would result in more formally specified models for better code generation. However, this would also reduce the flexibility of the modeller which is why we suggest that there should be two versions of the web interface.

Furthermore, integrating the conceptualization tool and the code snippets (used for translation to code), would increase the accessibility of MAIA to a great extent.

**Fully automatic translation of code.** In this research, we facilitated semiautomatic generation of code. By restricting the conceptualization process so that modellers provide only certain forms of input, it is possible to facilitate fully automatic translation of MAIA models to executable simulations. Since our goal was to minimize modelling efforts for non-programmers we did not put these restrictions on the MAIA meta-model because it would have increased the effort for learning MAIA. Nonetheless, if these restrictions are made, the trade off would be between learning a low level programming language such as Java or learning higher level concepts similar to MAIA and having executable code.

**Plug-ins and libraries.** Since the concepts in MAIA are general and abstract to cover a diversity of socio-technical systems, implementing more specific phenomena such as negotiation, cooperation and conflict resolution requires support. Building predefined libraries that provide institutions, entity actions and plans for such phenomena would be highly instrumental. Another possibility would be to implement learning algorithms that allow modelling adaptive agents. We expect that all these libraries can be coded using the basic concepts defined in the MAIA meta-model.

Besides these types of plug-ins it is also possible to build customized libraries and ontologies for specific domains such as electricity market, forestry management etc. These libraries ease the use of MAIA, enable reuse of models and broaden its application area.

**Building more socio-technical systems with MAIA.** Even though we have used a variety of case studies and examples to develop MAIA, more cases would help further improve the MAIA meta-model, the tool support and the translation of MAIA models to simulation code. While conducting these case studies, generic libraries and plug-ins may also be defined by extracting components that can be shared between different model.

**Facilitating institutional emergence.** The most important area for future development of this research is to facilitate institutional emergence. In the case studies we have developed with MAIA, institutions have been implemented and their effects on individuals and the system as a whole have been studied. The next step is to enable institutional change and creation of new institutions which may require complex algorithms. The computational requirements can be taken from MAS research and integrated into MAIA models.

Besides enabling institutional change and emergence, the detection and exploration of this aspect is another area of future research. Pattern recognition algorithms in artificial intelligence can be instrumental for this purpose.

Automating ABMS for Decision Support. The MAIA platform presented in this thesis provides a manual tool to make a conceptual model of a system, build simulations and analyse results. The main contributions of MAIA in this process are (1) enabling conceptualization from a high level modelling language, which can be understood by problem owners, and (2) facilitating semi-automatic code generation. In order to fully automate this process, a solution is to build intelligent agents that are able to make agent-based simulations. This implies that the agent would be able to build a simulation from a set of information provided by the problem owner (already facilitated by MAIA), analyse and interpret the results and give useful information back to the user. Therefore, the agent must be an intelligent entity who can make decisions about the conceptual model, the simulation, the results and the analysis.

# Appendices



# Evaluation Form for Usability and Usefulness of MAIA

As explained to the participants, the questions are about the concepts of MAIA not the software tool:

#### It is easy to understand the concepts in MAIA.

	1	2	3	4	5	6	7	
Completely Disagree	0	0	0		0		0	Completely Agree
It is easy to learn to	use	∍ M∕	AIA.					
	1	2	3	4	5	6	7	
Completely Disagree	0	0	0	0	0		0	Completely Agree
It is easy to become	ski	illfu	l in l	MAI	Α.			
	1	2	3	4	5	6	7	
Completely Disagree	0	0	0	0	0	0	0	Completely Agree
It is easy to build an	ag	ent-	bas	ed ı	nod	lel u	isin	g MAIA.
	1	2	3	4	5	6	7	

Completely Disagree 

Completely Agree

Completely Agree

Using MAIA is effective in the model development process. Does using MAIA make any difference in building agent-based models.

1 2 3 4 5 6 7

It is easy to collaborate with others to build an agent-based model using MAIA.

1 2 3 4 5 6 7

Completely Disagree 

Completely Agree

It is more efficient to build agent-based model with MAIA than other modelling tools. You can compare it with Java programming all the way to Netlogo and Repast.

1 2 3 4 5 6 7

Completely Disagree 

Completely Agree

Completely Agree

MAIA is applicable to other problems in my domain.

1 2 3 4 5 6 7

Completely Disagree 

Completely Agree

Completely Agree

Do you have any comments regarding the learning process of MAIA? Is there anything in particular that was difficult to learn?

Do you have any comments regarding the application of MAIA? Is there anything in particular that was difficult to apply t your case?

What are the benefits of using MAIA? name 3 if you can.

What are the drawbacks of using MAIA? name 3 if you can.

# **B** MAIA Syntax

The syntax of a modelling language is primarily associated with the well-formed formulas that define the constraints of that language. We explained these formulas for MAIA as an addition to the class diagram specifications in Chapter 3. The reason we primarily used these rules was to check the soundness of MAIA models.

At the implementation level, the syntax of MAIA was defined in XSD (XML Schema definition) format. XSD is used to define a set of rules to which an XML file must conform to. Since the definition of MAIA in ecore is stored in XML format, we use XSD specifications to define how the syntax of MAIA should be structured in the XML file. An example of a syntax rule in XSD format is illustrated for institutions in Listing 6. This rule shows that the deontic type is a required field for the definition of an institutional rule and that the 'or else' component of a rule is required and it is an institution itself. For each of the structures in MAIA we have generated an XSD file.

	<xsd:complextype_ecore:implements="constitutionalstruct:sanction" name="Rule"&gt;</xsd:complextype_ecore:implements="constitutionalstruct:sanction" 
2	< xsd : complexContent>
	<pre><xsd:extension <="" base="constitutionalStruct:InstitutionalStatement" pre=""></xsd:extension></pre>
	>
4	<pre><xsd:attribute ecore:unsettable="false" name="deonticType" type<br="">="constitutionalStruct:deonLogicType" use="required"/&gt;</xsd:attribute></pre>
	<pre><xsd:attribute <="" ecore:name="OrElse" ecore:reference="&lt;/pre&gt;&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;/th&gt;&lt;th&gt;constitutionalStruct:Sanction" name="OrElse" th="" use="required"></xsd:attribute></pre>
	>
6	<xsd:simpletype></xsd:simpletype>
	<xsd:list itemtype="xsd:anyURI"></xsd:list>
8	
10	
	</math xsd:complexContent>
2	

### Listing 6 – Syntax rules for an institution in XSD format.

# Evaluation Form for ABMS Platforms

The evaluation form consists of three tables. The first table evaluates the platform as a software tool. The second table evaluates the platform on how much it covers agentoriented aspects and the final table is about supports for simulation aspects. In the column on the right, a brief description is given about each concept. In the 'grade' column the respondent is requested to grade her tool following the grading system indicated below. If the feature is marked by a '\*', the grading scheme is different and is explained in the final column for that feature. A brief description can also be provided explaining why and how the platform supports that particular feature.

#### GRADING SCHEME

Generic scale point	Definition of Scale	Scale point mapping
Makes things worse	Cause Confusion. The way the feature is implemented makes it	-1
-	difficult to use and/or encouraged incorrect use of the feature	
No support	Fails to recognize it. The feature is not supported nor referred to	0
	in the user manual	
Little support	The feature is supported indirectly, for example by the use of	1
	other tool features in non-standard combinations.	
Some support	The feature appears explicitly in the feature list of the tools and	2
	user manual. However, some aspects of feature use are not	
	catered for.	
Strong support	The feature appears explicitly in the feature list of the tools and	3
	user manual. All aspects of the feature are covered but use	
	of the feature depends on the expertise of the user	
Very strong support	The feature appears explicitly in the feature list of the tools and	4
	user manual. All aspects of the feature are covered and the tool	
	provides tailored dialogue boxes to assist the user.	
Full support	The feature appears explicitly in the feature list of the tools and	5
	user manual. All aspects of the feature are covered and the	
	tool provides user scenarios to assist the user such as	
	"Wizards".	

## SOFTWARE ENGINEERING FEATURES

	Grade	Explanation	Description of this feature
MODELLING LANGUAGE AND	Notation		
Expressiveness			
Static + Dynamic aspects			Describes dynamic (e.g. data and
			control flows) and static aspects (e.g. structure and knowledge) of
			the system
Language Adequate and Expressive			Can any kind of social system be explained with the modeling
			language?
PRECISENESS		-	
Syntax			The tool provides an explicit, formal description of its syntax
Semantics			The tool provides an explicit,
			formal, description of its
Notation			The notation is clear and
			explicitly defined
MODEL COMPLEXITY MANAGEME	NT		The test summate the sheath of
Consistency check			consistency among the concepts
Refinement			The tool supports the refinement
			of the abstract concepts into more precise manageable concepts
Modularity			The tool enables Modularity of concepts
Hierarchical modeling			The tool enables Hierarchy of concepts
USER FRIENDLY			
Easy to use			Give measures/experiment results if any
Easy to learn			Give measures/experiment results if any
Easy to reuse			Give measures/experiment results if any
PROCESS (DOES THE TOOL SUPPO	RT THE FOLI	OWING SOFTWARE D	EVELOPMENT PHASES?)
Requirements Analysis			
Architectural /Detailed Design			
Implementation			
Testing and Debugging			~
Deployment			Run simulation
Methods of Transition between phases *			Is there Smooth or disjoint steps between the above mentioned
			phases in the development
Documentation			process?
PRAGMATICS			
MANAGEMENT DECISIONS			
Risk			The tool supports Risk
			assessment for the simulation
Role specification			Are the roles in the development
			process clearly defined? E.g., modeler, analyst, programmer

Scalable       Scalable       Scalable       Give measures/experiment results if any         Distributable / Team work       Distribution of tasks between people         Project Scheduling       Support for project scheduling, planning         EFFICIENCY       Time       How much time saving is your tool compared to other tools?         Give measures/experiment results if any       How efficient does your tool set other tools?         Resources       How efficient does your tool use freeworks of use of use fulness and efficient does your tool use fulness and efficiences; Give measures/experiment results if any         QUALITY       in terms of usefulness and effictiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Usefulness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or field to real work projects.		
Give measures/experiment results if any         Distribution of tasks between people         Project Scheduling         Bistribution of tasks between people         Project Scheduling         Bupport for project scheduling, planning         EFFICIENCY         Time         How much time saving is your tool compared to other tools?         Give measures/experiment results if any         Resources         How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY         Usefulness         Effectiveness         Give measures/experiment results if any         Usefulness         Sealable to larger simulations; Give measures/experiment results if any         Used by non-creators*         User/Stakeholder Involvement         Application*	Scalable	Scalable to larger simulations.
Distributable / Team work       Distribution of tasks between people         Project Scheduling       Support for project scheduling, planning         EFFICIENCY       Time         Time       How much time saving is your tool compared to other tools?         Give measures/experiment results if any       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Usefulness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to be number of real and complete projects the tool has been used for.		Give measures/experiment results
Distributable / Team work       Distribution of tasks between people         Project Scheduling       Support for project scheduling, planning         EFFICIENCY       How much time saving is your tool compared to other tools? Give measures/experiment results if any         Resources       How efficient does your tool use resources (e.g., computer, man power?) Give measures/experiment results if any         QUALITY       Usefulness         Effectiveness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know or?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to example simulations or applied to be number of real and complete projects the tool has been used for.		if any
Distribution of the number of the people       people         Project Scheduling       Support for project scheduling, planning         EFFICIENCY       How much time saving is your tool compared to other tools?         Give measures/experiment results if any       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Usefulness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations; and point of projects. * Indicate the number of real and complete projects.	Distributable / Team work	Distribution of tasks between
Project Scheduling       Support for project scheduling, planning         EFFICIENCY       How much time saving is your tool compared to other tools? Give measures/experiment results if any         Resources       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       Usefulness         Effectiveness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Sealable to larger simulations; Give measures/experiment results if any         Usefulness       Sealable to larger simulations; Give measures/experiment results if any         Usefulness       Sealable to larger simulations; Give measures/experiment results if any         Usefulness       Sealable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to real work projects.	District and the real work	people
EFFICIENCY       It is planning       planning         Time       How much time saving is your tool compared to other tools? Give measures/experiment results if any         Resources       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       usefulness         Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.	Project Scheduling	Support for project scheduling,
EFFICIENCY         Time       How much time saving is your tool compared to other tools? Give measures/experiment results if any         Resources       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       usefulness         Usefulness       in terms of usefulness and effectiveness, Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to real work projects.		planning
Time       How much time saving is your tool compared to other tools?         Give measures/experiment results if any       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       Usefulness         Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know o?         User/Stakeholder Involvement       Is the tool used to develop example simulations; or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.	EFFICIENCY	
Inite       tool compared to other tools?         Give measures/experiment results if any       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to real work projects.	Time	How much time saving is your
Give measures/experiment results         if any         Resources         How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY         Usefulness         in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness         Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*         User/Stakeholder Involvement         Application*         Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	Time	tool compared to other tools?
Resources       if any         QUALITY       In terms of usefulness and effectiveness; Give measures/experiment results if any         Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to real work projects.		Give measures/experiment results
Resources       How efficient does your tool use resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       Usefulness         in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.		if any
Resources       resources (e.g., computer, man power)? Give measures/experiment results if any         QUALITY       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know o?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	Pasouroos	How efficient does your tool use
QUALITY         Usefulness         in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness         Effectiveness         Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*         User/Stakeholder Involvement         Application*         Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	Resources	resources (e.g. computer man
QUALITY       measures/experiment results if any         QUALITY       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       Is the tool used to develop example simulations or applied to real work projects.		power)? Give
QUALITY         Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.		measures/experiment results if
QUALITY         Usefulness         Effectiveness         Give measures/experiment results if any         Effectiveness         Give measures/experiment results if fany         Used by non-creators*         User/Stakeholder Involvement         Application*         Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.		inclusives/experiment results in
Usefulness       in terms of usefulness and effectiveness; Give measures/experiment results if any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.		anv
Osefulness       In othe of using state         effectiveness       effectiveness, Give         measures/experiment results if       any         Effectiveness       Scalable to larger simulations;         Give measures/experiment results       if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	QUALITY	any
Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	QUALITY	any
Intesting experiment results in any         Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness	in terms of usefulness and
Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness	in terms of usefulness and effectiveness; Give
Effectiveness       Scalable to larger simulations; Give measures/experiment results if any         Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness	in terms of usefulness and effectiveness; Give measures/experiment results if
Used by non-creators*       What are the number of users to date that you know of?         User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         Application*       * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness	in terms of usefulness and effectiveness; Give measures/experiment results if any
Used by non-creators* What are the number of users to date that you know of? User/Stakeholder Involvement Application* Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations;
Used by non-creators*     What are the number of users to date that you know of?       User/Stakeholder Involvement     Is the tool used to develop example simulations or applied to real work projects.       * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results
User/Stakeholder Involvement     Is the tool used to develop example simulations or applied to real work projects.       * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any
User/Stakeholder Involvement       Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness Used by non-creators*	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any What are the number of users to
Application* Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness Used by non-creators*	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any What are the number of users to date that you know of?
example simulations or applied to real work projects.     * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness Used by non-creators* User/Stakeholder Involvement	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any What are the number of users to date that you know of?
real work projects. * Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness Used by non-creators* User/Stakeholder Involvement Application*	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any What are the number of users to date that you know of? Is the tool used to develop
* Indicate the number of real and complete projects the tool has been used for.	QUALITY Usefulness Effectiveness Used by non-creators* User/Stakeholder Involvement Application*	in terms of usefulness and effectiveness; Give measures/experiment results if any         Scalable to larger simulations; Give measures/experiment results if fany         What are the number of users to date that you know of?         Is the tool used to develop example simulations or applied to
complete projects the tool has been used for.	QUALITY         Usefulness         Effectiveness         Used by non-creators*         User/Stakeholder Involvement         Application*	in terms of usefulness and effectiveness; Give measures/experiment results if any       Scalable to larger simulations; Give measures/experiment results if any       What are the number of users to date that you know of?       Is the tool used to develop example simulations or applied to real work projects.
been used for.	QUALITY Usefulness Effectiveness Used by non-creators* User/Stakeholder Involvement Application*	in terms of usefulness and effectiveness; Give measures/experiment results if any Scalable to larger simulations; Give measures/experiment results if any What are the number of users to date that you know of? Is the tool used to develop example simulations or applied to real work projects. * Indicate the number of real and
	QUALITY Usefulness Effectiveness Used by non-creators* User/Stakeholder Involvement Application*	in terms of usefulness and effectiveness; Give measures/experiment results if any         Scalable to larger simulations; Give measures/experiment results if fany         What are the number of users to date that you know of?         Is the tool used to develop example simulations or applied to real work projects.         * Indicate the number of real and complete the tool has

## MAS FEATURES SUPPORTED BY YOUR TOOL

	Grade	Explanation	Description of this feature		
AGENT ATTRIBUTES					
Autonomy					
Mental Attitude			e.g., beliefs, desires, intentions, plans, personal values,		
Proactive					
Reactive					
AGENT ORGANIZATION	1				
Situated			Placed in an environment agents act accordingly		
Sociability			e.g., cooperate with other agents		
Rules			Follow rules		
Roles			take roles in the social system		
Tasks			perform tasks according to the specified organizational rules		
Distributed					
AGENT COMMUNICATION					
Protocols & Messages					
Concurrency					

## SOCIAL SIMULATION FEATURES SUPPORTED BY YOUR TOOL

CONCEPTUAL	
Environment	Simulation of the environment and artifacts besides agents
Multiple societies	As explicit organizational or relational entities
Ontologies	To integrate domain knowledge
ANALYSIS	
Data	
Event observation	
Visualization of results	
Detection and Exploration	
of unforeseen emergence	
IMPLEMENTATION	
Simulation Platform	Does the tool include a running simulation platform?
Scheduling	Does your tool schedule agents and tasks in the simulation?
Launch agents	Is your tool able to instantiate and launch agents in the simulation?
Intervene in behavior	Can the tool intervene in agent or system behavior during run time?
Static and dynamic	Static: verify concepts with
validation	stakeholders, dynamic: verify
	stakeholders.

## Bibliography

- Abbasy, A., van der Veen, R. and Hakvoort, R. (2011). Possible effects of balancing market integration on performance of the individual markets, *Energy Market* (*EEM*), 2011 8th International Conference on the European, IEEE, pp. 608–613. 99
- Afman, M. R., Chappin, E. J. L., Wander, J. and Dijkema, G. P. J. (2010). Agentbased model of transitions in consumer lighting, 3rd World Congress on Social Simulation, Kassel, Germany. 58, 59, 61
- Aldewereld, H. (2007). Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols, SIKS Dissertation Series 2007-10, Utrecht University. PhD Thesis. 135, 137
- AMP (2013). Agent modelling platform.
   URL: http://eclipse.org/amp x, 112, 115, 116, 117, 118, 119, 121
- Anderson, A. (1958). A reduction of deontic logic to alethic modal logic, Mind 67: 100–103. 137
- Andersson, K. (2006). Understanding decentralized forest governance: an application of the institutional analysis and development framework, Sustainability: Science Practice and Policy 2(1): 25–35. 54
- Andrighetto, G., Villatoro, D. and Conte, R. (2010). Norm internalization in artificial societies, AI Communications 23(4): 325–339. 142, 143
- Atkinson, C. and Kuhne, T. (2003). Model-driven development: a metamodeling foundation, Software, IEEE 20(5): 36–41. 27
- Axtell, R. (2001). Effects of interaction topology and activation regime in several multi-agent systems, *Multi-Agent-Based Simulation* pp. 33–48. 148
- Aylward, B. and González, A. (1998). Institutional arrangements for watershed management: A case study of arenal, costa rica. 54
- Azevedo, I. L., Morgan, G. and Morgan, F. (2009). The transition to solid-state lighting, *Proceedings of the IEEE* 97(3): 481–510. 59
- Bagni, R., Berchi, R. and Cariello, P. (2002). A comparison of simulation models applied to epidemics, *Journal of Artificial Societies and Social Simulation* 5(3). 99
- Baker, A. (2011). Simplicity, in E. N. Zalta (ed.), The Stanford Encyclopedia of

Philosophy, summer 2011 edn. 49

- Balci, O. (1997). Verification validation and accreditation of simulation models, Proceedings of the 29th conference on Winter simulation, IEEE Computer Society, pp. 135–141. 36
- Bankes, S. (2002). Agent-based modeling: A revolution?, Proceedings of the National Academy of Sciences of the United States of America 99(Suppl 3): 7199. 7
- Banks, J., Carson, J. S., Nelson, B. L. and Nicol, D. M. (2000). Discrete-Event System Simulation, 3rd edn, Prentice-Hall, Inc., Upper Saddle River, New Jersey. 19, 99
- Barreteau, O., Bousquet, F. and Attonaty, J.-M. (2001). Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to senegal river valley irrigated systems, *Journal of artificial societies and social simulation* **4**(2): 5. 46
- Becu, N., Neef, A., Schreinemachers, P. and Sangkapitux, C. (2008). Participatory computer simulation to support collective decision-making: Potential and limits of stakeholder involvement, *Land Use Policy* **25**(4): 498–509. 4
- Behdani, B. (2012). Handling Disruptions in Supply Chains: An Integrated Framework and an Agent-based Model, PhD thesis. 99
- Belnap, N. and Perloff, M. (1988). Seeing to it that: a canonical form for agentives, *Theoria* 54(3): 175–199. 136
- Benyon, D., Turner, P. and Turner, S. (2005). Designing interactive systems: People, activities, contexts, technologies, Addison-Wesley Longman. 49
- Bertoldi, P. and Atanasiu, B. (2007). Electricity consumption and efficiency trends in the enlarged european union – status report 2006. 59
- Boella, G. and Van Der Torre, L. (2003). Permissions and obligations in hierarchical normative systems, *Proceedings of the 9th international conference on Artificial intelligence and law*, ACM, pp. 109–118. 137
- Borshchev, A. and Filippov, A. (2004). Anylogicmulti-paradigm simulation for business, engineering and research, *The 6th IIE Annual Simulation Solutions Confer*ence. 150
- Bousquet, F., Bakam, I., Proton, H. and Le Page, C. (1998). Cormas: commonpool resources and multi-agent systems, *Tasks and Methods in Applied Artificial Intelligence* pp. 826–837. 150
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems 8(3): 203–236. 22, 149
- Broersen, J., Dignum, F., Dignum, V. and Meyer, J. (2004). Designing a deontic logic of deadlines, *Deontic Logic in Computer Science* pp. 43–56. 135

- Card, D., Mc Garry, F. and Page, G. (1987). Evaluating software engineering technologies, Software Engineering, IEEE Transactions on (7): 845–851. 47
- CEC (2009). Commission regulation (ec) no 244/2009 of 18 march 2009 implementing directive 2005/32/ec of the european parliament and of the council with regard to ecodesign requirements for non-directional household lamps, *Official Journal* of the European Union L076: 3–16. 59
- Cernuzzi, L. and Rossi, G. (2002). On the evaluation of agent oriented modeling methods, *Proceedings of Agent Oriented Methodology Workshop, Seattle.* 151
- Chappin, E. J. L. (2011). Simulations of Energy Transitions, PhD thesis, Delft University of Technology. ISBN: 978-90-79787-30-2. 58, 60, 99
- Chmieliauskas, A., Chappin, E. and Dijkema, G. (2012). Modeling socio-technical systems with agentspring, *Third International Engineering Systems Symposium* Design and Governance in Engineering Systems. 150
- Chu, T., Drogoul, A., Boucher, A. and Jucker, J. (2012). Towards a methodology for the participatory design of agent-based models, *Principles and Practice of Multi-Agent Systems* pp. 428–442. 4, 46
- Coleman, E. and Steed, B. (2009). Monitoring and sanctioning in the commons: An application to forestry, *Ecological Economics* **68**(7): 2106–2113. 54
- Coleman, J. (1986). Social theory, social research, and a theory of action, American journal of Sociology pp. 1309–1335. 17, 30, 170
- Conte, R., Edmonds, B., Moss, S. and Sawyer, R. (2001). Sociology and social theory in agent based social simulation: A symposium, *Computational & Mathematical Organization Theory* 7(3): 183–205. 4, 7, 19, 148, 170
- Conte, R., Gilbert, N. and Sichman, J. (1998). Mas and social simulation: A suitable commitment, *Multi-agent systems and agent-based simulation*, Springer, pp. 47– 53. 7
- Cossentino, M. (2005). From requirements to code with the passi methodology, Agent-oriented methodologies pp. 79–106. 149
- Coutinho, L., Sichman, J. and Boissier, O. (2009). Modelling dimensions for agent organizations. 22
- Crawford, S. and Ostrom, E. (1995). A grammar of institutions, American Political Science Review pp. 582–600. 14, 18, 33, 50, 52, 127, 131
- Cuesta, P., Gómez, A., González, J. and Rodríguez, F. (2003). A framework for evaluation of agent oriented methodologies, *Proceedings of the Conference of the* Spanish Association for Artificial Intelligence. 147, 151, 152, 154
- Curtis, S. (2005). Efficiency gains boost high-power LED performance, Compound Semiconductor 11(11): 27–30. 59
- Czarnecki, K. and Pietroszek, K. (2006). Verifying feature-based model templates

against well-formedness ocl constraints, Proceedings of the 5th international conference on Generative programming and component engineering, ACM, pp. 211– 220. 50

- Dam, K. and Winikoff, M. (2004). Comparing agent-oriented methodologies, Agent-Oriented Information Systems, Springer, pp. 78–93. 147, 151, 152, 153, 154
- De Korte, F. (2012). Exploring the potential of manure-based energy production in salland, Master's thesis, TU Delft. ix, x, 84, 86, 93, 94
- Dehnert, J. and Rittgen, P. (2001). Relaxed soundness of business processes, Advanced Information Systems Engineering, Springer, pp. 157–170. 48
- Desurvire, H., Caplan, M. and Toth, J. (2004). Using heuristics to evaluate the playability of games, CHI'04 extended abstracts on Human factors in computing systems, ACM, pp. 1509–1512. 47
- Dignum, F. and Kuiper, R. (1997). Combining dynamic deontic logic and temporal logic for the specification of deadlines, System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on, Vol. 5, IEEE, pp. 336–346. 135
- Dignum, V. (2004). A model for organizational interaction: based on agents, founded in logic, PhD thesis, University of Utrecht. 22, 32, 33, 36, 123, 135, 149
- Dignum, V. and Aldewereld, H. (2010). Operetta: Organization-oriented development environment, Proc. of LADS'2010@Mallow. 22
- Dignum, V., Dignum, F., Osinga, S. and Hofstede, G. (2010). Normative, cultural and cognitive aspects of modelling policies, *Simulation Conference (WSC)*, *Proceedings of the 2010 Winter*, IEEE, pp. 720–732. 148
- Dignum, V., Tranier, J. and Dignum, F. (2010). Simulation of intermediation using rich cognitive agents, Simulation Modelling Practice and Theory 18(10): 1526– 1536. 148
- Drossopoulou, S., Valkevych, T. and Eisenbach, S. (2000). Java type soundness revisited, *Department of Computing, Imperial College London, Tech. Rep* **9**. 48
- Dunin-Keplicz, B. and Verbrugge, R. (2002). Collective intentions, Fundamenta Informaticae 51(3): 271–295. 138, 142
- Eclipse (2013). Eclipse.org home. URL: http://www.eclipse.org 113
- Emerson, E. (1990). Temporal and modal logic, in J. van Leeuwen (ed.), Handbook of Theoretical Computer Science, Vol. B, MIT Press, pp. 955–1072. 135
- Engelfriet, J. (1996). Minimal temporal epistemic logic, Notre Dame Journal of Formal Logic 37(2): 233-259. 135
- Epstein, J. (2006). Generative social science: Studies in agent-based computational modeling, Princeton Univ Pr. 30, 169
- Esteva, M., de la Cruz, D. and Sierra, C. (2002). Islander: an electronic institu-

tions editor, Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, ACM, pp. 1045–1052. 147, 150

- Esteva, M., Rodriguez-Aguilar, J., Sierra, C., Garcia, P. and Arcos, J. (2001). On the formal specification of electronic institutions, *Agent mediated electronic commerce* pp. 126–147. 96, 123
- Evans, A. (1998). Reasoning with uml class diagrams, Industrial Strength Formal Specification Techniques, 1998. Proceedings. 2nd IEEE Workshop on, IEEE, pp. 102–113. 50
- Evans, A., France, R., Lano, K. and Rumpe, B. (1999). The uml as a formal modeling notation, *The Unified Modeling Language. UML*»98: Beyond the Notation pp. 514–514. 50
- Ferber, J., Michel, F. and Baez, J. (2005). Agre: Integrating environments with organizations, *Environments for Multi-agent Systems* pp. 48–56. 22, 32, 149
- Forrester, J. (1961). Industrial dynamics, Vol. 5, MIT press Cambridge, MA. 19
- Frank, R., Dabbaghian, V., Reid, A., Singh, S., Cinnamon, J. and Brantingham, P. (2011). Power of criminal attractors: modeling the pull of activity nodes, *Journal* of Artificial Societies and Social Simulation 14(1): 6. 99
- Fuentes-Fernández, R., Hassan, S., Pavón, J., Galán, J. and López-Paredes, A. (2012). Metamodels for role-driven agent-based modelling, *Computational & Mathematical Organization Theory* pp. 1–22. 113
- Fuxman, A., Kazhamiakin, R., Pistore, M. and Roveri, M. (2003). Formal tropos: language and semantics, University of Trento and IRST 55. 123
- García-Magariño, I., Gómez-Rodríguez, A., Gómez-Sanz, J. and González-Moreno, J. (2009). Ingenias-scrum development process for multi-agent development, *In*ternational Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), Springer, pp. 108–117. 156, 157
- Garro, A., Parisi, F. and Russo, W. (2012). A process based on the model-driven architecture to enable the definition of platform-independent simulation models, *Simulation and Modeling Methodologies, Technologies and Applications* pp. 113– 129. 112, 150
- Garro, A. and Russo, W. (2010). easyabms: A domain-expert oriented methodology for agent-based modeling and simulation, Simulation Modelling Practice and Theory 18(10): 1453–1467. 28, 149, 157
- Gendre, M. F. (2003). Two centuries of electric light source innovations. URL: http://www.einlightred.tue.nl/lightsources/history/light\_history.pdf 59
- Ghorbani, A. (2012). A software tool for conceptualizing agent-based models, 14th European Agent Systems Summer School Paper Session (EASSS 2012). 38
- Ghorbani, A., Aldewereld, H., Dignum, V. and Dijkema, G. (2012). Applying the maia platform to model a manure-based energy system in the netherlands, *The*

eighth conference of the European Social Simulation Association (ESSA). 84

- Ghorbani, A., Aldewereld, H., Dignum, V. and Noriega, P. (2012). Dutch eat at 5: 30 pm: Shared strategies in agent systems, *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*. 135
- Ghorbani, A., Aldewereld, H., Dignum, V. and Noriega, P. (2013). Shared strategies in artificial agent societies., *Coordination, Organizations, Intitutions, and Norms* in Agent Systems VIII, LNCS, **7756**. 135, 169
- Ghorbani, A., Bots, P., Dignum, V. and Dijkema, G. (2013). Maia: A framework for developing agent-based social simulations, *Journal of Artificial Societies and Social Simulation* 16(2). 28
- Ghorbani, A. and Dignum, V. (2013). Agent-based simulation for complex social systems: Support for the developer, *Proceedings of the European Conference on Complex Systems*, Complexity, Springer. 113
- Ghorbani, A., Dignum, V. and Dijkema, G. (2012). An analysis and design framework for agent-based social simulation, Advanced Agent Technology, Lecture Notes in Artificial Intelligence, Springer (ISBN 978-3-642-27215-8): 96–112. 28
- Ghorbani, A., Dignum, V., Sheoratan, S. and Dijkema, G. (2011). Applying the maia methodology to model the informal ewaste recycling sector, *The seventh conference of the European Social Simulation Association (ESSA)*. 72
- Ghorbani, A., Ligtvoet, A., Nikolić, I. and Dijkema, G. (2010). Using institutional frameworks to conceptualize agent-based models of socio-technical systems, in S. Stepney, P. H. Welch, P. S. Andrews and A. T. Sampson (eds), Proceedings of the 2010 Workshop on Complex Systems Modelling and Simulation, Odense, Denmark. 12
- Giddens, A. (1984). The constitution of society: Outline of the theory of structuration, Univ of California. 16, 17, 18, 28, 29, 32, 33, 34, 35, 36
- Giddens, A. and Turner, J. (1988). Social theory today, Stanford Univ Pr. 28
- Gilbert, N. (2004). Agent-based social simulation: dealing with complexity, The Complex Systems Network of Excellence 9(25): 1–14. 3, 4, 7
- Gilbert, N., Maltby, S. and Asakawa, T. (2002). Participatory simulations for developing scenarios in environmental resource management, *Third workshop on agentbased simulation*, SCS-European Publishing House Ghent,, Belgium, pp. 67–72. 46
- Gómez-Sanz, J., Fernández, C. and Arroyo, J. (2010). Model driven development and simulations with the ingenias agent framework, *Simulation Modelling Practice* and Theory 18(10): 1468–1482. 156
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S., Huse, G. et al. (2006). A standard protocol for describing individual-based and agent-based models, *Ecological modelling* 198(1-

2): 115-126. 149

- Grosz, B. and Kraus, S. (1996). Collaborative plans for complex group action, Artificial Intelligence 86(2): 269–357. 142, 143
- Grosz, B. and Sidner, C. (1988). Plans for discourse, *Technical report*, DTIC Document. 142, 143
- Guitouni, A. and Martel, J. (1998). Tentative guidelines to help choosing an appropriate mcda method, *European Journal of Operational Research* **109**(2): 501–521. 103
- Gutknecht, O. and Ferber, J. (2000). Madkit: A generic multi-agent platform, Proceedings of the fourth international conference on Autonomous agents, ACM, pp. 78–79. 150
- Guyot, P. and Honiden, S. (2006). Agent-based participatory simulations: Merging multi-agent systems and role-playing games, *Journal of Artificial Societies and Social Simulation* 9(4). 46
- Ha, N., Agusa, T., Ramu, K., Tu, N., Murata, S., Bulbule, K., Parthasaraty, P., Takahashi, S., Subramanian, A. and Tanabe, S. (2009). Contamination by trace elements at e-waste recycling sites in Bangalore, India, *Chemosphere* 76(1): 9–15. 31, 73
- Hannoun, M., Boissier, O., Sichman, J. and Sayettat, C. (2000). Moise: An organizational model for multi-agent systems, Advances in Artificial Intelligence pp. 156–165. 22, 149
- Hassan, S., Fuentes-Fernández, R., Galán, J., López-Paredes, A. and Pavón, J. (2009). Reducing the modeling gap: On the use of metamodels in agent-based simulation, 6th Conference of the European Social Simulation Association (ESSA 2009), pp. 1–13. 28, 149
- Hausman, J. A. (1979). Individual discount rates and the purchase and utilization of energy-using durables, *The Bell Journal of Economics* 10(1): 33–54. URL: http://www.jstor.org/stable/3003318 59
- Heath, B., Hill, R. and 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. 19, 155
- Hedström, P. and Swedberg, R. (1996). Social mechanisms, *Acta Sociologica* **39**(3): 281–308. 32
- Hedstrom, P. and Swedberg, R. (1998). Social mechanisms: An introductory essay, Social mechanisms: An analytical approach to social theory pp. 1–31. ix, 17, 18, 27, 29, 30, 33, 36
- Henderson-Sellers, B. and Giorgini, P. (2005). Agent-oriented methodologies, IGI Global. 151
- Hesari, S., Mashayekhi, H. and Ramsin, R. (2010). Towards a general framework for

evaluating software development methodologies, 2010 34th Annual IEEE Computer Software and Applications Conference, IEEE, pp. 208–217. 147, 151, 152, 153, 154

- Hindriks, K., Jonker, C., Kraus, S., Lin, R. and Tykhonov, D. (2009). Genius: negotiation environment for heterogeneous agents, *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume* 2, International Foundation for Autonomous Agents and Multiagent Systems, pp. 1397–1398. 70
- Hodgson, G. (1988). Economics and institutions, Journal of Economic Issues, Citeseer. 96
- Hodgson, G. and Calatrava, J. (2006). What are institutions, Journal of Economic Issues 40(1): 1. 7, 13
- Hodgson, G. and Knudsen, T. (2004). The complex evolution of a simple traffic convention: the functions and implications of habit, *Journal of Economic Behavior & Organization* 54(1): 19–47. 133
- Holonyak, J. N. (2005). From transistors to lasers and light-emitting diodes, Materials Research Society Bulletin 30: 509–515. 59
- Huysman, M. (1994). Waste picking as a survival strategy for women in indian cities, Environment and Urbanization 6(2): 155–174. 73
- Iba, T., Matsuzawa, Y. and Aoyama, N. (2004). From conceptual models to simulation models: Model driven development of agent-based simulations, 9th Workshop on Economics and Heterogeneous Interacting Agents. 28, 149
- Iglesias, C., Garijo, M., González, J. and Velasco, J. (1998). Analysis and design of multiagent systems using mas-commonkads, *Intelligent Agents IV Agent Theories*, *Architectures, and Languages* pp. 313–327. 149
- Imperial, M. and Yandle, T. (2005). Taking institutions seriously: using the iad framework to analyze fisheries policy, *Society and Natural Resources* 18(6): 493– 509. 54
- Janssen, M., Alessa, L., Barton, M., Bergin, S. and Lee, A. (2008). Towards a community framework for agent-based modelling, *Journal of Artificial Societies* and Social Simulation 11(2): 6. 27, 28
- Jones, R. (1965). The structure of simple general equilibrium models, The Journal of Political Economy 73(6): 557–572. 15
- Kent, S. (2002). Model driven engineering, Integrated Formal Methods, Springer, pp. 286–298. 20, 22
- Kiser, L. and Ostrom, E. (2000). The three worlds of action: A metatheoretical synthesis of institutional approaches, *Polycentric Games and Institutions* 1: 56– 88. 50
- Kitchenham, B. (1996). Evaluating software engineering methods and tool part

1: The evaluation context and evaluation methods, ACM SIGSOFT Software Engineering Notes **21**(1): 11–14. 147, 151

- Kitchenham, B., Linkman, S. and Law, D. (1997). DESMET: a methodology for evaluating software engineering methods and tools, *Computing & Control Engin*eering Journal 8(3): 120–126. 47, 48, 50, 57, 147, 151, 156
- Klijn, E. and Koppenjan, J. (2006). Institutional design, Public management review 8(1): 141–160. 13
- Klugl, F., Herrler, R. and Fehler, M. (2006). Sesam: implementation of agent-based simulation using visual programming, *International Conference on Autonomous* Agents: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, Vol. 8, pp. 1439–1440. 150
- Kooreman, P. (1996). Individual discounting, energy conservation, and household demand for lighting, *Resource and Energy Economics* 18(1): 103–114. 59
- Koppenjan, J. and Groenewegen, J. (2005). Institutional design for complex technological systems, *International Journal of Technology*, *Policy and Management* 5(3): 240–257. 13
- Kostadinov, F., Holm, S., Steubing, B., Thees, O. and Lemm, R. (2012). Simulation of a swiss wood fuel and roundwood market: An explorative study in agent-based modeling, *submitted: Policy and Economics*. ix, xi, 67, 68, 70, 71
- Kostadinov, F. and Steubing, B. (2011). An agent-based model of an energy wood market in a swiss region, The seventh conference of the European Social Simulation Association (ESSA). 67
- Koua, E., MacEachren, A. and Kraak, M. (2006). Evaluating the usability of visualization methods in an exploratory geovisualization environment, *International Journal of Geographical Information Science* 20(4): 425–448. 47
- Lam, P., Lam, S., Lam, J. and McNaught, C. (2009). Usability and usefulness of ebooks on ppcs: How students opinions vary over time, Australasian Journal of Educational Technology 25(1): 30–44. 49
- Landauer, T. (1996). The trouble with computers: Usefulness, usability, and productivity, Taylor & Francis. 57
- Lawis, P., Mark, K., Thomas, D. and Courtheyn, T. (2001). A formal process for evaluating cots software products. 47
- Ligtvoet, A., Ghorbani, A. and Chappin, E. (2011). A methodology for agentbased modeling using institutional analysis applied to consumer lighting, Agent Technologies for Energy Systems, Tenth international conference on Autonomous Agents and Multi Agent Systems (AAMAS), Taipei, Taiwan. 60
- Luck, M., McBurney, P. and Preist, C. (2003). Agent technology: enabling next generation computing (a roadmap for agent based computing), AgentLink/University of Southampton. 19, 20, 155

- Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K. and Balan, G. (2005). Mason: A multiagent simulation environment, *Simulation* 81(7): 517–527. 150
- Macal, C. and North, M. (2010). Tutorial on agent-based modelling and simulation, Journal of Simulation 4(3): 151–162. 19, 155
- Makinson, D. and Van Der Torre, L. (2000). Input/output logics, Journal of Philosophical Logic 29(4): 383–408. 135
- Mantelzorg (2013). Mantel zorg. URL: http://www.rijksoverheid.nl/onderwerpen/mantelzorg 101
- Marietto, M., David, N., Sichman, J. and Coelho, H. (2003). Requirements analysis of agent-based simulation platforms: State of the art and new prospects, *Multi-Agent-Based Simulation II* pp. 183–201. 147, 148, 154, 155
- Marsh, H. and Hau, K. (1996). Assessing goodness of fit: Is parsimony always desirable?, The Journal of Experimental Educational pp. 364–390. 49
- Martinot, E. and Borg, N. (1998). Energy-efficient lighting programs. experience and lessons from eight countries, *Energy Policy* 26(14): 1071–1081. 59
- Menanteau, P. and Lefebvre, H. (2000). Competing technologies and the diffusion of innovations: The emergence of energy-efficient lamps in the residential sector, *Research Policy* 29(3): 375–389. 59, 60
- Meyer, J.-J. C. and van der Hoek, W. (1995). Epistemic Logic for AI and Computer Science, Cambridge University Press. 135, 140
- Mills, E. (1993). Efficient lighting programs in europe: Cost effectiveness, consumer response, and market dynamics, *Energy* 18(2): IN1,131–144. 59
- Minar, N. (1996). The swarm simulation system: A toolkit for building multi-agent simulations, Santa Fe Institute Santa Fe, NM. 20, 150
- Moody, D. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, *Data & Knowledge Engineering* 55(3): 243–276. 47, 48, 50, 151
- Morgenstern, P. and De Groot, G. (2010). Bio-energiecentrales: Inventariserend Onderzoek Naar Milieuaspecten Bij Diverse Energieopwekkingstechnieken Met Behulp Van Biomassa, RIVM. 85
- Moss, S. (2002). Policy analysis from first principles, Proceedings of the National Academy of Sciences of the United States of America 99(Suppl 3): 7267. 20
- Nikolic, I. (2009). Co-Evolutionary Method For Modelling Large Scale Socio-Technical Systems Evolution, PhD thesis. 4
- Nikolic, I. and Ghorbani, A. (2011). A method for developing agent-based models of socio-technical systems, Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on, IEEE, pp. 44–49. 27
- North, D. (1990). Institutions, institutional change and economic performance, Cam-

bridge university press. 7, 13

- North, M., Collier, N. and 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. 20, 27, 44, 147, 150, 155
- Oakerson, R. (1992). Analyzing the commons: A framework, Making the commons work: Theory, practice and policy (oo): 41–59. 13
- Okada, I. (2011). An agent-based model of sustainable corporate social responsibility activities, *Journal of Artificial Societies and Social Simulation* 14(3): 4. 99
- Okuyama, F., Bordini, R. and da Rocha Costa, A. (2007). Spatially distributed normative objects, *Coordination, organizations, institutions, and norms in agent* systems II pp. 133–146. 142
- OMG (2013). Model driven architecture. URL: http://www.omg.org/mda/ 111, 112
- Ostrom, E. (1991). Crafting institutions for self-governing irrigation systems. 13, 33
- Ostrom, E. (1999). Institutional rational choice: An assessment of the institutional analysis and development framework, *Theories of the policy process* pp. 35–71. 49
- Ostrom, E. (2005). Understanding institutional diversity, Princeton Univ Pr. ix, 7, 12, 13, 28, 32, 36, 43, 51, 52, 53, 131, 132, 133, 135, 137, 141, 142
- Ostrom, E., Gardner, R. and Walker, J. (1994). Rules, games, and common-pool resources, Univ of Michigan Pr. ix, 7, 12, 14, 28, 29, 34, 37, 49, 50, 143
- Padgham, L. and Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents, Agent-Oriented Software Engineering III pp. 174–185. 149
- Parker, M. (2001). What is ascape and why should you care, Journal of Artificial Societies and Social Simulation 4(1): 5. 27, 150
- Pavon, J., Gomez-Sanz, J. and Fuentes, R. (2005). The ingenias methodology and tools, Agent-oriented methodologies pp. 236–276. 22, 147, 149, 156
- Pavón, J., Gómez-Sanz, J. and Fuentes, R. (2006). Model driven development of multi-agent systems, *Model Driven Architecture-Foundations and Applications*, Springer, pp. 284–298. 113, 121
- Pavon, J., Sansores, C. and Gomez-Sanz, J. (2008). Modelling and simulation of social systems with ingenias, *International Journal of Agent-Oriented Software Engineering* 2(2): 196–221. 4, 78, 99, 100, 101, 106, 123, 149, 150, 156
- Pettit, P. (1996). Institutional Design and Rational Choice, Cambridge University Press, pp. 54–89. 13
- Peyton, H. (1993). The evolution of conventions, *Econometrica* **61**(1): 57–84. 142
- Railsback, S. F., Lytinen, S. L. and Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations, *Simulation* 82(9): 609– 623. 4, 9, 165

- Ramanath, A. and Gilbert, N. (2004). The design of participatory agent-based social simulations, Journal of Artificial Societies and Social Simulation 7(4). 4, 5, 6, 20, 36, 46, 55, 99, 165
- Rao, A. and Georgeff, M. (1995). Bdi agents: From theory to practice, Proceedings of the first international conference on multi-agent systems (ICMAS-95), San Francisco, pp. 312–319. 72
- Reiser, R. and Dick, W. (1990). Evaluating instructional software, Educational Technology Research and Development 38(3): 43–50. 47
- Saaty, T. (2008). Decision making with the analytic hierarchy process, International Journal of Services Sciences 1(1): 83–98. 67, 72
- Sansores, C. and Pavón, J. (2005). Agent-based simulation replication: A model driven architecture approach, MICAI 2005: Advances in Artificial Intelligence pp. 244–253. 28, 156
- Schank, R., Abelson, R. et al. (1977). Scripts, plans, goals and understanding: An inquiry into human knowledge structures, Vol. 2, Lawrence Erlbaum Associates Nueva Jersey. 133
- Scharpf, F. (1997). Games real actors play: Actor-centered institutionalism in policy research, Westview Press. 3, 7, 13, 15, 18, 28, 29, 31, 32, 33, 34, 102
- Schluep, M. and Programme, U. N. E. (2009). Recycling: from E-waste to Resources, UNEP and United Nations University. 73
- Schrauwen, N. (2012). Innovativeness of innovation, Master's thesis, TU Delft. 167
- Sears, A. and Jacko, J. (2007). The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, CRC. 49
- Segars, A. and Grover, V. (1993). Re-examining perceived ease of use and usefulness, MIS quarterly 17(4): 517–525. 47, 49
- Shehory, O. and Sturm, A. (2001). Evaluation of modeling techniques for agentbased systems, *Proceedings of the fifth international conference on Autonomous* agents, ACM, pp. 624–631. 147, 151, 152, 154
- Sheoratan, S. (2011). Agent-based modelling of the e-waste recycling sector in bangalore, Master's thesis, Delft University of Technology, Leiden University. ix, xi, 73, 81, 82
- Sivo, S. and Willson, V. (1998). Is parsimony always desirable? identifying the correct model for a longitudinal panel data set, *The Journal of experimental education* 66(3): 249–255. 48, 49
- Sloman, A. and Logan, B. (1999). Building cognitively rich agents using the sim\_agent toolkit, Communications of the ACM 42(3): 71-ff. 148
- Smajgl, A., Leitch, A. and Lynam, T. (2009). Outback institutions: an application of the institutional analysis and development (iad) framework to four case studies

in australias outback, Desert Knowledge . 50, 52, 54

- Sofia Panagiotidi and Javier Vázquez-Salceda (2011). Normative Planning: Semantics and Implementation, 13th International Workshop on Coordination, Organizations, Institutions and Norms in Agent Systems (COIN@WI-IAT), Lyon, France. 139
- Stahl, T. T. and Völter, M. (2006). Model-driven software development, John Wiley & Sons New York. 20, 21
- Steubing, B., Kostadinov, F., Ghorbani, A., Wager, P., Zaha, R., Thees, C. and Ludwig, C. (2011). Agent-based modeling of a woodfuel market factors that affect the availability of woodfuel, *ISIE'11*, *Berkeley*, US. 67
- Steubing, B., Zah, R., Waeger, P. and Ludwig, C. (2010). Bioenergy in switzerland: Assessing the domestic sustainable biomass potential, *Renewable and Sustainable Energy Reviews* 14(8): 2256–2265. 66
- Sturm, A. and Shehory, O. (2004). A framework for evaluating agent-oriented methodologies, Agent-Oriented Information Systems, Springer, pp. 94–109. 147, 151
- Sudeikat, J., Braubach, L., Pokahr, A. and Lamersdorf, W. (2005). Evaluation of agent–oriented software methodologies–examination of the gap between modeling and platform, Agent-Oriented Software Engineering V pp. 126–141. 151
- Terna, P. (1998). Simulation tools for social scientists: Building agent based models with swarm, Journal of Artificial Societies and Social Simulation 1(2): 1–12. 155
- Thomas, C. and Fischer, G. (1996). Using agents to improve the usability and usefulness of the world-wide web, *Fifth International Conference on User Modeling*, pp. 5–12. 49
- Tisue, S. (2004). Netlogo: Design and implementation of a multi-agent modeling environment, In Proceedings of Agent 2004, Citeseer. 20, 27, 150, 155
- Togridou, A., Hovardas, T. and Pantis, J. (2006). Factors shaping implementation of protected area management decisions: a case study of the zakynthos national marine park, *Environmental Conservation* **33**(3): 233–243. 54
- Tsakonas, G. and Papatheodorou, C. (2006). Analysing and evaluating usefulness and usability in electronic information services, *Journal of Information Science* **32**(5): 400–419. 49
- Tsakonas, G. and Papatheodorou, C. (2008). Exploring usefulness and usability in the evaluation of open access digital libraries, *Information Processing & Management* 44(3): 1234–1250. 49
- van der Aalst, W. (1997). Verification of workflow nets, Application and Theory of Petri Nets 1997 pp. 407–426. 48
- van der Veen, R., Abbasy, A. and Hakvoort, R. (2012). Agent-based analysis of the impact of the imbalance pricing mechanism on market behaviour in electricity balancing markets, *Energy Economics*. 99

- Villatoro, D., Sen, S. and Sabater-Mir, J. (2010). Of social norms and sanctioning: A game theoretical overview, *International Journal of Agent Technologies and Systems (IJATS)* 2(1): 1–15. 142
- Von Wright, G. (1951). Deontic logic, Mind 60(237): 1-15. 135
- Weiss, G. (1999). Multiagent systems: a modern approach to distributed artificial intelligence, The MIT press. 19
- Wijngaards, N., Overeinder, B., van Steen, M. and Brazier, F. (2002). Supporting internet-scale multi-agent systems, *Data & Knowledge Engineering* 41(2): 229– 245. 150
- Williamson, O. (1998). Transaction cost economics: how it works; where it is headed, The Economist 146(1): 23–58. 7, 13
- Winograd, T., Bennett, J., De Young, L. and Hartfield, B. (1996). Bringing design to software, ACM Press New York, NY. 27
- Wooldridge, M. and Jennings, N. (1995). Intelligent agents: Theory and practice, Knowledge engineering review 10(2): 115–152. 19
- Wright, A., Felleisen, M. et al. (1994). A syntactic approach to type soundness, Information and computation 115(1): 38–94. 48
- Wright, G. (1977). Norm and action: a logical enquiry, Routledge & Kegan Paul. 132
- Yu, E. and Cysneiros, L. (2002). Agent-oriented methodologies-towards a challenge exemplar, Proc of the 4 Intl. Bi-Conference Workshop on AOIS, Toronto. 151

## Summary

Socio-technical systems consist of many heterogeneous decision making entities and technological artefacts. These systems are governed through public policy that unravels in a multi-scale institutional context, which ranges from norms and values to technical standards. For example, to influence consumer behaviour towards more energy saving habits, various policies and instruments can be employed, such as taxation on energy consuming light bulbs or a subsidy on the purchase of energy efficient but expensive LED lamps. New standards for lamp-fittings can be introduced, light bulbs can be banned, and LED-lamps may be promoted. Designing effective policies essentially requires insights into socio-technical systems.

Simulation is an exploratory approach to gain insights into socio-technical systems and investigate the possible outcomes of policy interventions. Simulations allow the identification of desired and undesired social and technological behaviours in a system and enable the exploration of scenarios that do not exist in the actual system. Agent-based modelling and simulation (ABMS) is particularly suitable for studying socio-technical systems because it provides a natural representation of these systems. ABMS allows heterogeneous decision making entities, the so called agents to interact, causing the structure and global behaviour of the simulated system to emerge. Thus, ABMS facilitates the understanding of how certain behaviours of or events upon system elements result in emergent system outcomes.

In agent-based systems, agents commonly represent people, companies, governments, technological artefacts and other 'self-contained' entities. One limitation of common representations of agent-based systems is that agents are modelled to act and react in an environment which lacks social structure, an institutional context. Social structures, such as norms and culture, however, affect decision making entities in socio-technical systems and are in turn affected by them.

In conventional ABMS, these structures are either not considered at all or are implicitly modelled as part of the agents. Since social structures stand at a higher level than decision making entities in socio-technical systems, the primary consequence of modelling them within agents is that we effectively do not model the possibility of change of these structures, and we cannot simulate and observe how new structures would emerge and existing ones evolve or perish.

Presently, there exist not only the aforementioned conceptual limitation but also practical drawbacks for ABMS. First, compared to other simulations, agent-based models are relatively complex to build, and they require substantial programming knowledge. However, the actual users of simulations are social scientists and policy makers who may have little familiarity with computational tools. Second, with current ABMS tools and methods, it is rather difficult to involve various parties in
the simulation development process. However, communication with domain experts and problem owners to bring in their knowledge and expertise, is a major requirement for building adequate simulations.

ABMS is an insightful tool for studying socio-technical systems. However, to really understand and link various levels of behaviour in these systems and increase the usability of ABMS, we need to overcome the aforementioned conceptual and practical limitations of this approach. This motivated us to formulate our research question as follows:

How can we build social structures in agent-based models of socio-technical systems?

A related question is: *How can we increase the utility of ABMS for problem owners?* 

We addressed these questions in three steps:

- To give social structure to agent-based models of socio-technical systems, we designed a modelling framework.
- To facilitate the use of ABMS by modellers with different levels of expertise, including ones who are less familiar with computer programming or simulation, we provided tool support for the modelling framework.
- To enable collaborative model development, we enable participation, starting from the early conceptualization phase of the simulation development process.

The modelling framework is designed to *integrate* social structures into agentbased models. This implies that it defines social structures for these models *and* it explicates the link between these structures and other components of the system. To build such framework we identified all the components of a socio-technical system that are influenced by social structures and vice versa. Since our objective is to underpin and facilitate building models and simulations of complex socio-technical systems from solid foundations, we had to unravel the complex relationships between the various components of such systems.

To improve our understanding of social structures, we used institutional analysis which is commonly applied to study social systems. Institutions are sets of rules that structure social behaviour and interaction. Therefore, they can be considered as the building blocks of social structure. The institutional analysis and development framework (IAD) by the Nobel laureate Ostrom, provides a clear definition for institutions and specifies the connection between this social concept and physical, individual and operational aspects of a social system. Where IAD did not provide sufficient details, we extended the framework with three other social theories and frameworks, namely: structuration theory, actor centred institutionalism and social mechanism. The result of this process is an agent-based modelling framework called MAIA: Modelling Agent systems based on Institutional Analysis.

Using the IAD framework also brought us one step closer to reaching the second objective of this research, to make ABMS more accessible to users with limited aptitude towards software and programming. Since MAIA uses the IAD language, the concepts defined in it are more understandable than computational concepts to users who are social scientists and policy analysts. We built a user interface for MAIA which further facilitates its use. Finally, we provided development guidelines, defined protocols to transform MAIA models to simulations, and developed software modules that facilitate semi-automatic translation of models into executable simulation code.

Our third consideration was to facilitate participatory model development which we demonstrate through four case studies: consumer lighting transitions, woodfuel market, e-waste recycling network and bio-gas investments. These case studies demonstrated that on the one hand, the modeller can present a MAIA model to domain experts and problem owners to conceptually verify the model and make sure that her understanding of the system is in line with the system and the problem at hand. On the other hand, that MAIA can be used to separate simulation tasks and involve people with different areas of expertise. The modeller conceptualizes the socio-technical system, relying on her knowledge of the system and without having concerns about simulation development. A programmer with computational expertise can use the conceptualized model to build an executable simulation, without requiring domain knowledge and only relying on her computational expertise.

The overall outcome of this research is the MAIA modelling platform which consists of (1) a modelling *language* (i.e., framework) for conceptualizing agent-based models of socio-technical systems, (2) a web tool that facilitates the conceptualization process, (3) transformation protocols that enable the translation of models into simulations and (4) software modules that facilitate semi-automatic translation of models to simulation code.

MAIA has been evaluated in several stages. First, since we claim that our framework is a formalization and extension of IAD, we have verified its compliance with this framework. Second, since one of our goals was to widen the scope of ABMS and bring this simulation approach within the reach of inexperienced modellers, we have tested and confirmed its usability and usefulness for such users through the case studies mentioned above. The case study work further revealed that the concepts in the framework are sufficient to decompose and model each case study, that the conceptualization process can be executed by inexperienced users and that it is possible to develop a simulation from a conceptual model. Third, we have developed a comprehensive framework to evaluate MAIA as a simulation platform and have compared it with other ABMS platforms. This final evaluation confirmed that (1) the MAIA platform is conceptually more comprehensive than other simulation platforms, (2) it follows software development guidelines and (3) it takes agent-oriented artificial intelligence on board to an extent that is suitable for building simulations.

In conclusion, this research advances ABMS as a simulation approach to study socio-technical systems. The MAIA framework, as a modelling language, provides a template to conceptualize a system, suggesting components of a typical system including social structure that may be relevant to consider in the model and simulation. However, the level of complexity and the selection of components depend on the type of problem addressed and on the perception of the modeller. This reflects the flexibility of MAIA in building simulations.

This research also has implications for the policy analysis domain. With MAIA,

this domain benefits from rich and structured simulations of the system under study, in addition to having a test bed to explore the effects of alternative policies. MAIA allows explicit and structured policy modelling, which enables the comparison between policy alternatives with respect to their effects on the socio-technical system.

Our final contribution is to multi-agent systems literature by providing a structure to model institutions in artificial societies, and enabling the emergence and evolution of social structures.

## Samenvatting

Socio-technische systemen worden gevormd door een heterogene verzameling van mensen, organisaties en technologische artefacten.

Deze systemen worden gestuurd door overheden via beleid dat zich afspeelt in een meerlaagse institutionele context die variërt van normen en waarden tot technische standaarden. Bijvoorbeeld, om consumentengedrag te beïnvloeden en meer energie-besparende gewoonten te bevorderen kunnen verschillende typen beleid en beleidsinstrumenten worden ingezet, zoals bijvoorbeeld een belasting op energieverslindende gloeilampen of juist subsidies op de aanschaf van LED-lampen. Standaarden voor LED-verlichtingsarmaturen kunnen worden geïntroduceerd, gloeilampen kunnen worden verboden, LED-lampen kunnen worden gepromoot.

Het ontwerpen van *effectief* beleid vereist in de kern inzicht in de werking van socio-technische systemen: wat maakt dat LED-lampen de aloude peertjes doen verdwijnen? Simulatie maakt het mogelijk om dit inzicht te verkrijgen en om de mogelijke uitkomsten van beleidsinterventies te onderzoeken. Simulatie is een verkennende benadering die ons in staat stelt gewenste en ongewenste kenmerken van systemen te identificeren die hun oorsprong vinden in sociaal en technisch gedrag. Ze staat ons toe scenario's te beproeven die in de echte wereld niet kunnen of mogen voorkomen.

Agent-geBaseerd Modelleren en Simuleren (ABMS) is een natuurlijk representatie van deze systemen, en ze is als zodanig daarom bijzonder geschikt om sociotechnische systemen te bestuderen. ABMS laat de elementen die samen het systeem vormen, de zogenaamde agenten, communiceren en interacteren. Uit deze interactie ontstaan de structuren en het gedrag van het gesimuleerde systeem. Daarmee geeft ABMS inzicht in hoe het gedrag van de elementen van een systeem leidt tot de emergentie van structuur en gedrag van het systeem als geheel.

In agent-gebaseerde systemen vertegenwoordigen agenten mensen, bedrijven, overheden, technologische artefacten en andere zelfstandige entiteiten. Een beperking van de gebruikelijke representatie van agent-gebaseerde systemen is dat de agenten gemodelleerd zijn alsof ze handelen in een omgeving zonder sociale structuren of institutionele context. Het zijn echter juist deze sociale structuren, zoals normen en cultuur, die de besluitvorming in socio-technische systemen beïnvloeden en die op hun beurt gevormd worden door de besluitvorming.

In conventionele ABMS worden deze structuren in het geheel niet of slechts impliciet gemodelleerd als onderdeel van de agenten. In socio-technische systemen vormen sociale structuren echter als het ware de laag die het gedrag van de beslissers bepaalt. Het modelleren en simuleren van deze laag binnen de agenten maakt het moeilijk, zo niet onmogelijk, om juist de verandering in deze structuren te kunnen simuleren, en zo te observeren hoe nieuwe sociale structuren ontstaan en hoe bestaande sociale structuren evolueren of verdwijnen.

Momenteel kent ABMS niet alleen methodologische beperkingen maar ook praktische. Ten eerste zijn agent-gebaseerde modellen relatief complex om te bouwen in vergelijking met andere typen simulaties; ontwikkeling van een ABMS vereist kennis van en het vermogen tot programmeren. De feitelijke gebruikers van de simulaties zijn echter veelal sociale wetenschappers en beleidsmakers, die niet altijd voldoende kennis van en ervaring met informatica, programmeren en software-ontwikkeling zullen hebben. Ten tweede is het met de thans beschikbare ABMS methoden, technieken en hulpmiddelen lastig om verschillende partijen effectief te betrekken bij de ontwikkeling van een simulatie. Communicatie die er toe leidt dat de kennis en ervaring van specialisten, experts en probleemeigenaars effectief wordt ontsloten en gebruikt is cruciaal voor het bouwen van betekenisvolle simulaties.

ABMS is een hulpmiddel bij het bestuderen van socio-technische systemen die veel inzicht oplevert. Echter, om de verschillende niveaus van gedrag in deze systemen echt te kunnen begrijpen en te verbinden, en om de bruikbaarheid van ABMS te verhogen is het zaak haar conceptuele, methodologische en praktische beperkingen op te heffen.

Dit motiveerde ons om in ons onderzoek de volgende vraag centraal te stellen:

Hoe kunnen we sociale structuren bouwen in agent-gebaseerde modellen van sociotechnische systemen?

Een verwante vraag is:

Hoe kunnen we het nut van ABMS voor probleemeigenaren verhogen?

We hebben deze vragen in drie stappen geaddresseerd:

- Om agent-gebaseerde modellen van socio-technische systemen te verrijken met sociale structuren hebben we een modelleringskader ontworpen.
- Ter vergemakkelijking van het gebruik van ABMS voor modelbouwers met verschillende niveaus van expertise, met inbegrip van degenen die minder bekend zijn met het programmeren van computers of simulatie, hebben we een applicatie ontwikkeld ter ondersteuning van het gebruik van het modelleringskader.
- Om alle betrokkenen bij modelontwikkeling samen te laten werken, ondersteunen we participatief ontwikkelen vanaf de vroege conceptualiseringsfase van het simulatieontwikkelproces.

Het modelleringskader is ontworpen om sociale structuren te vervlechten in agentgebaseerde modellen. Dit betekent dat het kader de sociale structuren voor deze modellen definieert *en* dat het kader expliciet het verband legt tussen deze structuren en andere componenten van het systeem. Om een dergelijke kader te kunnen opzetten hebben we alle componenten van een socio-technisch systeem geïdentificeerd die worden beïnvloed door sociale structuren en vice versa. Vervolgens was het nodig de kluwen van relaties tussen de verschillende onderdelen van dergelijke systemen te ontwarren, zodat we ze konden formaliseren om een basis te verkrijgen voor het naspeelbaar ontwikkelen van modellen en simulaties van complexe socio-technische systemen.

Om ons begrip van sociale structuren te vergroten hebben we institutionele analyse gebruikt, een theorie die vaak wordt ingezet om inzicht in sociale systemen te verkrijgen. Instituties zijn sets van regels die sociaal gedrag en interactie structureren. Daarom kunnen ze worden beschouwd als de bouwstenen van een sociale structuur. Het 'Institutional Analysis and Development' (IAD) kader van de Nobelprijswinnaar Ostrom geeft een duidelijke definitie van instituties en geeft het verband tussen dit sociale concept en de fysieke, individuele en operationele aspecten van een sociaal systeem. Daarom hebben we IAD gebruikt als basis voor het bouwen ons agent-gebaseerde modelleerkader. Waar IAD niet voldoende detail bood hebben we het uitgebreid met drie andere sociale theorieën en kaders, te weten: structuratietheorie, actor-gecentreerd institutionalisme en sociaal mechanisme. Het resultaat van dit proces is een agent-gebaseerd modelleerkader genaamd MAIA: 'Modelling Agent systems with Institutional Analysis'.

Door het IAD-kader te gebruiken kwamen we ook een stap dichter bij de tweede doelstelling van dit onderzoek: ABMS meer toegankelijk maken voor gebruikers die minder belangstelling hebben voor software en programmeren of daarin ervaring ontberen. Omdat MAIA de IAD-taal gebruikt, zijn de in MAIA gebruikte concepten goed te hanteren door sociale wetenschappers, bestuurskundigen en beleidsanalisten. Om de drempel voor gebruikers nog verder te verlagen hebben we een gebruikersinterface voor MAIA gebouwd. Daarnaast hebben we een handleiding en richtlijnen ontwikkeld en protocollen gedefinieerd om MAIA modellen om te zetten naar simulaties. Ten slotte hebben we softwaremodules ontwikkeld voor de semi-automatische vertaling van de MAIA-modellen naar uitvoerbare code.

Ons derde doel was om participatieve modelontwikkeling te faciliteren. Dit illustreren we aan de hand van vier case studies: de transitie van gloeilamp naar LEDverlichting, de hout-brandstofmarkt, het e-waste recycling netwerk en ten slotte investeringen in bio-gas. Deze case studies hebben laten zien dat enerzijds de modelbouwer een MAIA-model kan presenteren aan domeinexperts en probleemeigenaars om het model conceptueel te verifiëren en te verzekeren dat haar begrip van het systeem in overeenstemming is met het systeem en het model het voorliggende probleem betreft; anderzijds kan MAIA worden gebruikt om een scheiding aan te brengen in taken zodat mensen met verschillende expertise c.q. uit verschillende vakgebieden effectief in het ontwikkelproces betrokken kunnen zijn. De modelbouwer conceptualiseert het socio-technische systeem, waarbij ze vertrouwt op haar kennis van het systeem, zonder zich te hoeven bekommeren om het ontwikkelen en programmeren van een simulatie. Een ervaren programmeur kan vervolgens het conceptuele model gebruiken als vertrekpunt voor de bouw van een uitvoerbare simulatie, terwijl ze geen domeinkennis bezit, maar juist vertrouwt op haar kennis van software-ontwikkeling en programmeren.

Het resultaat van dit onderzoek is het MAIA-modelleerplatform dat bestaat uit (1) een modelleer*taal* (d.w.z. kader) voor het conceptualiseren van agent-gebaseerde modellen van socio-technische systemen, (2) een webtool die het proces van concep-

tualisering faciliteert, (3) transformatieprotocollen voor de vertaling van modellen in simulaties en (4) softwaremodules voor de semi-automatische vertaling van modellen tot simulatie-code.

MAIA is geëvalueerd in verschillende fasen. Om te beginnen hebben we bekeken of MAIA aan de regels van het IAD-kader voldoet omdat we claimen dat ons raamwerk een formalisering en uitbreiding van IAD is. In de tweede plaats hadden we ons onder meer ten doel gesteld om de reikwijdte van ABMS te vergroten en om deze simulatieaanpak binnen bereik van relatief onervaren modelbouwers te brengen. Daarom hebben we het nut en de bruikbaarheid van MAIA voor dergelijke gebruikers getest door de genoemde case studies uit te voeren, om zo eveneens te kunnen vaststellen of de concepten die het ontwikkelde kader biedt voldoende zijn om het systeem - i.c. elk van de case studies - te kunnen ontleden en te modelleren, en of het conceptualiseringsproces kan worden uitgevoerd door relatief onervaren gebruikers en of het mogelijk is een simulatie te ontwikkelen met alléén een conceptueel model als basis. Ten derde hebben we een breed kader ontwikkeld om MAIA te evalueren als een simulatieplatform en hebben we MAIA vergeleken met andere ABMS platforms. Deze eindevaluatie bevestigt dat (1) het MAIA platform conceptueel uitgebreider is dan andere simulatie-platforms, (2) dat MAIA de formele richtlijnen voor software-ontwikkeling volgt en (3) dat MAIA agent-georiënteerde kunstmatige intelligentie aan boord haalt op een manier die geschikt is voor de bouw van simulaties.

Tot slot, dit onderzoek zet een stap in de ontwikkeling van ABMs als een simulatieaanpak voor de studie van socio-technische systemen. Het MAIA kader biedt, als modelleertaal, een template voor het conceptualiseren van een systeem. Ze suggereert de componenten van een typisch systeem inclusief de sociale structuur die wellicht relevant is voor het model en de simulatie. Echter, de complexiteit van het model en de keuze van haar componenten is afhankelijk van het type probleem en van de perceptie van de modelbouwer. Dit weerspiegelt de flexibiliteit van MAIA bij de ontwikkeling van simulaties.

Dit onderzoek heeft ook implicaties voor het domein van de beleidsanalyse. Met MAIA kan de beleidsanalyse profiteren van rijke en gestructureerde simulaties van het bestudeerde systeem. Tegelijkertijd vormen deze een test bed dat geschikt is om de effecten van alternatief beleid te verkennen. MAIA maakt expliciete en gestructureerde modellen van beleid mogelijk, en daarmee de vergelijking tussen beleidsalternatieven en hun effecten op het socio-technisch systeem.

Ten slotte draagt ons werk bij aan de literatuur van multi-agent systemen, doordat het een structuur biedt voor het modelleren van instituties in een kunstmatige samenleving, en omdat het de mogelijkheid biedt sociale structuren te laten ontstaan en te evolueren *in silico*.

## Curriculum vitae

Amineh Ghorbani was born on May 10, 1983 in Isfahan, Iran. Having completed her high school education in Mathematics and Physics, she was admitted to the Computer Engineering bachelor program at Isfahan University of Technology in 2001. During her bachelors she became especially interested in software engineering, taking this specialization as her major. Later on, as she gained experience outside university as an IT consultant, she decided to focus more on the applied side of software engineering and therefore studied for the national university exam for the IT master program. Being ranked 12th in the country, she was admitted to University of Tehran to continue her M.Sc. studies in 2007. Multi-agent systems was the next topic she became interested in. Therefore, she decided to further continue her studies as a PhD researcher in an area where she could apply a combination of software engineering and multi-agent systems research.

In September 2009, Amineh started her PhD research at the Energy and Industry Section, Delft University of Technology. By combining software engineering and multi-agent systems research with yet another topic of interest, simulation, she continued her PhD research on developing an agent-based simulation tool from software engineering principles. She made great use of the knowledge in the faculty of Technology, Policy and Management to build a social science backbone for her tool.

Amineh enjoyed attending numerous conferences and taking ideas and perspectives on her topic of research on board as she came across them. During her PhD research, Amineh organized several workshops on agent-based modelling for policy analysis and arranged a special issue for the AI & Society journal on the same topic.

This CV was last updated on May 7, 2013.

# NGInfra PhD thesis series on infrastructures

- 1. Strategic behavior and regulatory styles in the Netherlands energy industry. *Martijn Kuit*, 2002, Delft University of Technology, the Netherlands.
- Securing the public interest in electricity generation markets The myths of the invisible hand and the copper plate. *Laurens de Vries*, 2004, Delft University of Technology, the Netherlands.
- 3. Quality of service routing in the internet Theory, complexity and algorithms. *Fernando Kuipers*, 2004, Delft University of Technology, the Netherlands.
- 4. The role of power exchanges for the creation of a single European electricity market – Market design and market regulation. *François Boisseleau*, 2004, Delft University of Technology, the Netherlands and University of Paris IX Dauphine, France.
- 5. The ecology of metals. *Ewoud Verhoef*, 2004, Delft University of Technology, the Netherlands.
- 6. MEDUSA Survivable information security in critical infrastructures. Semir Daskapan, 2005, Delft University of Technology, the Netherlands.
- Transport infrastructure slot allocation. Kaspar Koolstra, 2005, Delft University of Technology, the Netherlands.
- 8. Understanding open source communities An organizational perspective. *Ruben van Wendel de Joode*, 2005, Delft University of Technology, the Netherlands.
- Regulating beyond price Integrated price-quality regulation for electricity distribution networks.
   Viren Ajodhia, 2006, Delft University of Technology, the Netherlands.
- Networked reliability Institutional fragmentation and the reliability of service provision in critical infrastructures. *Mark de Bruijne*, 2006, Delft University of Technology, the Netherlands.
- 11. Regional regulation as a new form of telecom sector governance The interactions with technological socio-economic systems and market performance. *Andrew Barendse*, 2006, Delft University of Technology, the Netherlands.

- The internet bubble The impact on the development path of the telecommunications sector.
   Wolter Lemstra, 2006, Delft University of Technology, the Netherlands.
- 13. Multi-agent model predictive control with applications to power networks. *Rudy Negenborn*, 2007, Delft University of Technology, the Netherlands.
- 14. Dynamic bi-level optimal toll design approach for dynamic traffic networks. *Dusica Joksimović*, 2007, Delft University of Technology, the Netherlands.
- Intertwining uncertainty analysis and decision-making about drinking water infrastructure. Machtelt Meijer, 2007, Delft University of Technology, the Netherlands.
- The new EU approach to sector regulation in the network infrastructure industries. *Richard Cawley*, 2007, Delft University of Technology, the Netherlands.
- A functional legal design for reliable electricity supply How technology affects law.
   Hamilcar Knops, 2008, Delft University of Technology, the Netherlands and Leiden University, the Netherlands.
- 18. Improving real-time train dispatching Models, algorithms and applications. *Andrea D'Ariano*, 2008, Delft University of Technology, the Netherlands.
- Exploratory modeling and analysis A promising method to deal with deep uncertainty. *Datu Buyung Agusdinata*, 2008, Delft University of Technology, the Netherlands.
- 20. Characterization of complex networks Application to robustness analysis. Almerima Jamaković, 2008, Delft University of Technology, the Netherlands.
- Shedding light on the black hole The roll-out of broadband access networks by private operators. Marieke Fijnvandraat, 2008, Delft University of Technology, the Netherlands.
- 22. On Stackelberg and inverse Stackelberg games & their applications in the optimal toll design problem, the energy markets liberalization problem, and in the theory of incentives. *Kateřina Staňková*, 2009, Delft University of Technology, the Netherlands.
- 23. On the conceptual design of large-scale process & energy infrastructure systems

   Integrating flexibility, reliability, availability, maintainability and economics performance metrics.
   Austine Ajah, 2009, Delft University of Technology, the Netherlands.
- Comprehensive models for security analysis of critical infrastructure as complex systems. *Fei Xue*, 2009, Politecnico di Torino, Torino, Italy.

- Towards a single European electricity market A structured approach for regulatory mode decision-making. *Hanneke de Jong*, 2009, Delft University of Technology, the Netherlands.
- Co-evolutionary process for modelling large scale socio-technical systems evolution. Iaor Nikolić, 2009. Delft University of Technology, the Netherlands.
- Regulation in splendid isolation A framework to promote effective and efficient performance of the electricity industry in small isolated monopoly systems. Steven Martina, 2009, Delft University of Technology, the Netherlands.
- 28. Reliability-based dynamic network design with stochastic networks. *Hao Li*, 2009, Delft University of Technology, the Netherlands.
- Competing Public Values Coping strategies in heavily regulated utility industries. Bauke Steenhuisen, 2009, Delft University of Technology, the Netherlands.
- Innovative contracting practices in the road sector Cross-national lessons in dealing with opportunistic behaviour. *Mónica Altamirano*, 2009, Delft University of Technology, the Netherlands.
- Reliability in urban public transport network assessment and design. Shahram Tahmasseby, 2009, Delft University of Technology, the Netherlands.
- Capturing socio-technical systems with agent-based modelling. Koen van Dam, 2009, Delft University of Technology, the Netherlands.
- Road incidents and network dynamics Effects on driving behaviour and congestion.
   Victor L. Knoop, 2009, Delft University of Technology, the Netherlands.
- 34. Governing mobile service innovation in co-evolving value networks. Mark de Reuver, 2009, Delft University of Technology, the Netherlands.
- 35. Modelling risk control measures in railways. Jaap van den Top, 2009, Delft University of Technology, the Netherlands.
- 36. Smart Heat and Power Utilizing the Flexibility of Micro Cogeneration. Michiel Houwing, 2010, Delft University of Technology, the Netherlands.
- Architecture-Driven Integration of Modeling Languages for the Design of Software-Intensive Systems. *Michel dos Santos Soares*, 2010, Delft University of Technology, the Netherlands.
- Modernization of electricity networks Exploring the interrelations between institutions and technology. *Martijn Jonker*, 2010, Delft University of Technology, the Netherlands.

- Experiencing Complexity A gaming approach for understanding infrastructure systems. *Geertje Bekebrede*, 2010, Delft University of Technology, the Netherlands.
- 40. Epidemics in Networks Modeling, Optimization and Security Games. Jasmina Omic, 2010, Delft University of Technology, the Netherlands.
- Designing Robust Road Networks A general method applied to the Netherlands. Maaike Snelder, 2010, Delft University of Technology, the Netherlands.
- 42. Simulating Energy Transitions. *Emile Chappin*, 2011, Delft University of Technology, the Netherlands.
- 43. De ingeslagen weg. Een dynamisch onderzoek naar de dynamiek van de uitbesteding van onderhoud in de civiele infrastructuur. Rob Schoenmaker, 2011, Delft University of Technology, the Netherlands.
- Safety Management and Risk Modelling in Aviation: the challenge of quantifying management influences. *Pei-Hui Lin*, 2011, Delft University of Technology, the Netherlands.
- 45. Transportation modelling for large-scale evacuations. *Adam J. Pel*, 2011, Delft University of Technology, the Netherlands.
- 46. Clearing the Road for ISA Implementation? Applying Adaptive Policymaking for the Implementation of Intelligent Speed Adaptation. *Jan-Willem van der Pas*, 2011, Delft University of Technology, the Netherlands.
- Designing Multinational Electricity Balancing Markets. Reinier van der Veen, 2012, Delft University of Technology, the Netherlands.
- Understanding Social-technical Change A System-Network-Agent Approach. Catherine M. Chiong Meza, 2012, Delft University of Technology, the Netherlands.
- 49. National design and multi-national integration of balancing markets. *Alireza Abbasy*, 2012, Delft University of Technology, the Netherlands.
- 50. Regulation of gas infrastructure expansion. Jeroen de Joode, 2012, Delft University of Technology, the Netherlands.
- 51. Governance Structures of Free/Open Source Software Development. Examining the role of modular product design as a governance mechanism in the FreeBSD Project. *George Dafermos*, 2012, Delft University of Technology, the Netherlands.
- 52. Making Sense of Open Data From Raw Data to Actionable Insight. Chris B. Davis, 2012, Delft University of Technology, the Netherlands.

- Intermodal Barge Transport: Network Design, Nodes and Competitiveness. *Rob Konings*, 2009, Delft University of Technology, Trail Research School, the Netherlands.
- 54. Handling Disruptions in Supply Chains: An integrated Framework and an Agent-based Model. Behzad Behdani, 2013. Delft University of Technology, the Netherlands.
- 55. Images of cooperation a methodological exploration in energy networks. Andreas Ligtvoet, 2013, Delft University of Technology, the Netherlands.
- Robustness and Optimization of Complex Networks: Spectral analysis, Modeling and Algorithms.
   Dajie Liu, 2013, Delft University of Technology, The Netherlands
- 57. Wegen door Brussel: Staatssteun en publieke belangen in de vervoersector. Nienke Saanen, 2013, Delft University of Technology, The Netherlands
- The Flexible Port.
   Poonam Taneja, 2013, Delft University of Technology, The Netherlands
- 59. Transit-Oriented Development in China; How can it be planned in complex urban systems? Rui Mu, 2013, Delft University of Technology, The Netherlands
- Cross Culture Work: Practices of Collaboration in the Panama Canal Expansion Program. Karen Smits, 2013, VU University Amsterdam, The Netherlands
- Structuring Socio-technical Complexity: Modelling Agent systems using Institutional Analysis. *Amineh Ghorbani*, 2013, Delft University of Technology, The Netherlands

### Structuring Socio-technical Complexity Modelling Agent Systems Using Institutional Analysis

Socio-technical systems consist of many heterogeneous decision making entities and technological artefacts. These systems are governed through public policy that unravels in a multi-scale institutional context. For example, to influence consumer behaviour towards more energy saving habits, various policies and instruments can be employed such as taxation on energy consuming light bulbs or subsidy on the purchase of energy efficient but expensive LED lamps. Designing effective policies essentially requires insights into socio-technical systems which can be gained through agent-based modeling and simulation.

This research builds on a combination of artificial intelligence, software engineering and institutional analysis. MAIA is introduced as a modeling framework that integrates social structures into agent-based models of socio-technical systems. Besides supporting inexperienced modellers, MAIA also acts as a tool to support participatory model development.

#### The Next Generation Infrastructures Foundation

represents an international consortium of knowledge institutions, market players and governmental bodies, which joined forces to cope with the challenges faced by today's and tomorrow's infrastructure systems. The consortium cuts across infrastructure sectors, across discplinary borders and across national borders, as infrastructure systems themselves do. With the strong participation of practitioners in a concerted knowledge effort with social and engineering scientists, the Foundation seeks to ensure the conditions for utilization of the research results by infrastructure policy makers, regulators and the infrastructure industries. **WWW.nginfra.nl** 



NEXT GENERATION INFRASTRUCTURES FOUNDATION



