CoModels, engineering dynamic compositions of coupled models to support the simulation of complex systems

Quang-Nghi Huynh

DOCTORAL THESIS
PIERRE AND MARIE CURIE UNIVERSITY
Speciality

Informatics - Complex Systems Modeling
Doctoral School EDITE de Paris

CoModels: engineering dynamic compositions of coupled models to support the simulation of complex systems

By Quang-Nghi HUYNH

Presented and publicly defended on 05 December 2016

Examining committee:

Alexis DROGOUL, Directeur de Recherches, IRD (Thesis Director)
Jean-Daniel ZUCKER, Directeur de Recherches, IRD (Co-supervisor)
Bernard POTTIER, Professeur , UBO (Reviewer)
Fabien MICHEL, Maître de Conférences, LIRMM (Reviewer)
Nathalie CORSON, Maître de Conférences, UFR (Examiner)
Christian VINCENOT, Professeur, Université de Kyoto (Examiner)
Minh-Thu TRAN-NGUYEN, Maître de Conférences, CICT (Examiner)

0.1 Acknowledgement

Firstly, I would like to express my sincere gratitude to my two advisors, Alexis Drogoul and Huynh Xuan Hiep, and to my co-encadrant, Jean-Daniel Zucker, for their continuous support, patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in doing research and writing this thesis. I would like to thank them as well for all the activities they organized, such as the training programs, coding camps, which brought me to the world of research. I would like to thank the organizers, professors, and secretaries of the PDI program (Programme Doctoral International) for allowing me to participate to the international network. I would like to thank Ms. Patricia Zizzo for her help for administrative points at the UPMC. A special thanks to Jean-Daniel Zucker, Edith Perrier, Christophe Cambier, Nicolas Marilleau, Philippe Caillou, Vo Duc An, Patrick Taillandier, Benoit Gaudou and Kathy Baumont for their advices, lessons and supports. I would like to thank my professors at UPMC (Paris), CTU (Cantho), IRD (Bondy) and USTH (Hanoi). I would like to say a really big thank you to Dr. Lai Hien Phuong who has taken a lot of times to review the manuscript.

In addition, I would like to thank the members of my thesis committee, for their insightful comments, and hard questions. My sincere thanks also go to Vo Huynh Tram, Phan Phuong Lan and to all my colleagues at Can Tho University who gave me the best conditions for doing research during these almost four years. I thank all my fellow lab mates at DREAM Team, GAMA developers and CIT for the stimulating discussions, Truong Xuan Viet, Nguyen Nhi Gia

Vinh, Truong Minh Thai, Truong Chi Quang for their supports, suggestions for my research, and for all the fun we have had together. I also thank my friends, Arnaud Grignard, Nguyen Huu Tri, Tran Nguyen Minh Thu, Nguyen Ngoc Doanh for their friendship and helpfulness.

Last but not least, I would like to thank my family: my parents HUYNH Van Sai and HUYNH Thi Nguyet, for their constant support in my life, and my wife DUONG Kim Ngan, for all her love and understanding. I also thank my sister for her dedicated help during my study, HUYNH Nguyet Thanh.

0.2 Abstract

Computer modeling, in line with the evolution of software engineering, is evolving from the design of integral models to that of integrated models. The complexity of today's modeling systems, especially when they are intended to be used for decision-making in complex socio-environmental contexts, pledges for the use of flexible modeling techniques and support tools. As a matter of fact, it is more and more common to integrate, into a same model, different sub-models which are defined at different scales of time and space, and which can be expressed in different formalisms. It can be due to the fact that they are legacy models or that they support different points of view from different domains of expertise.

Integrated modeling approaches (multi-simulation, multimodeling, etc.) have however proven challenging in practice. The first challenge deals with the technical aspects of coupling different computational or mathematical components. The second challenge lies in the alignment of the semantics of these components so that their integration does make sense, which is particularly critical in pluridisciplinary models. A number of approaches have been proposed in the last 20 years to address these challenges. However, for different reasons described in this manuscript, none of them is really suitable to our context .

We propose in this thesis an alternate approach, called co-modeling, which borrows concepts and tools from agent-based modeling, agent-oriented software engineering and multimodel ecologies. Simply speaking, a co-model can be defined as a multi-agent system of models and datasets. Each model or

dataset is represented by one or several agents interacting with one another within the context of a larger representation of their — potentially dynamic — environment. The proposed approach does not aim at providing a general solution to the two challenges above, but at providing a framework in which modelers can easily implement their solution or test different coupling solutions.

The proposed approach is fully implemented within the GAMA agent-based modeling platform. Its advantages are shown in terms of flexibility, composability and reusability in a number of case studies. The first case study is the dynamic coupling of equation-based and agent-based models to obtain "switching" models dynamically. The second one is the design of a complex integrated model where three formalisms and four modeling approaches have been successfully coupled.

Table of contents

0.1 Acknowledgement 3

0.2 Abstract 5

1 Introduction 11

1.1 Context 11

1.2 Motivations and Research questions 16

2 STATE OF THE ART 20

2.1 Why coupling models ? 20

2.1.1 Model 20

2.1.2 Models Coupling 21

2.1.3 Advantages of coupling 22

2.2 Different kinds of coupling 26

2.2.1 Weak coupling 27

2.2.2 Strong coupling 29

2.3 Challenges of model coupling 31

2.3.1 Reusability 32

2.3.2 Scalability 33

2.3.3 Expressivity 34

2.3.4 Flexibility 35
2.4 Existing solutions in modeling/simulation 35
2.5 Conclusions 45

3 CO-MODEL: AN INFRASTRUCTURE FOR SUPPORTING THE DYNAMICAL COUPLING OF HETEROGENEOUS MODELS 47
3.1 Introduction 48
3.2 Basic concepts 48
3.2.1 Co-model 49
3.2.2 Micro-model 50
3.3 Integration in the GAMA platform 51
3.3.1 Why GAMA? 51
3.3.2 Implementation 52
3.3.3 Portability 54
3.4 Example of use (syntaxes) 55
3.4.1 Importation 55
3.4.2 Instantiation 55
3.4.3 Execution 56
3.5 End of chapter 57

4 Demonstration and usage 59
4.1 Objective 59
4.2 Toy model description 60
4.3 Toy model implementation 60

4.3.1 The animal-resource model . . . 61
4.3.2 The prey-predator model . . . 63
4.4 Co-modeling . . . 65
4.4.1 Step by step co-modeling the prey-predator co-model . . 65
Step 1 . . . 65
Step 2 . . . 65
Step 3 . . . 66
Step 4 . . . 66
4.5 Discussion . . . 67
4.5.1 Reusability . . . 67
4.5.2 Expressivity . . . 69
4.5.3 Scalability . . . 71
4.5.4 Flexibility . . . 71

5 Dynamic choice of the best representation of a phenomenon 72
5.1 Objective . . . 73
5.2 Modeling context . . . 74
5.3 Definition of micro-models . . . 76
5.4 Transformation of the Switch model into a co-model . . . 81
5.4.1 Dynamics of co-model . . . 82
5.5 Conclusion . . . 83

6 Incremental design of a complex integrated model 85
6.1 Objective . . . 85
6.2 Modeling context . . . 86

6.3 Definition of micro-models 90
6.3.1 Farmers behaviors model (M_F) 90
6.3.2 Salinity intrusion model (M_S) 91
6.3.3 Parcels model (M_P) 92
6.3.4 Economical model (M_E) 93
6.3.5 Farmers relationships model (M_N) 95
6.3.6 Summary on the micro-models 96
6.4 Impact of environmental factors on farmers' decisions 97
6.4.1 Implementation 98
Step 1 98
Step 2 98
Step 3 98
Step 4 99
6.5 Coupling of Farmer and Socio-Economy factors 100
6.5.1 Implementation 100
Step 1 100
Step 2+3+4 100
6.6 Coupling environmental, social and economic models 102
6.6.1 Implementation 103
Step 1+2+3+4 103
6.6.2 Experimentation 104
6.7 Conclusion 106

7 Conclusion **108**

Chapter 1

Introduction

Multi-modelisation is not different with the modelisation. It does not need a new language to describe the coupling of models.

1.1 Context

Complex systems span the whole spectrum from life sciences and medicine, physics, chemistry and engineering, social, economic, and cognitive sciences. Research in the domain of complex systems requires a truly interdisciplinary approach that crosses traditional disciplines. In a complex system, interactions of individual components produce emergent functionalities, not found at the individual level, and one of the challenges of contemporary science is to understand this phenomenon. Modeling and simulating such systems in all their complexity requires however a multi-modeling approach, as the interactions can "belong" to different disciplines. The models produced by a multi-modeling approach are called "coupled" or "integrated". They encompass multiple sub-models, some of them, called "legacy models", having been designed to answer a specific question in a different context. Designing coupled models is a challenge nowadays for anyone working on complex systems, especially because the notion of “coupling” can represent any link between the different sub-models: dynamic interactions and feedbacks, static or dynamic compositions or combinations of these models in various frameworks.

Different definitions of complex system could be found in the literature:

- "A system comprised of a (usually large) number of (usually strongly) interacting entities, processes, or agents, the understanding of which requires the development, or the use of, new scientific tools, nonlinear models, out-of equilibrium descriptions and computer simulations" [1]
- "A system that can be analyzed into many components having relatively many relations among them, so that the behavior of each component depends on the behavior of others”. Simon, Herbert A. (1973). The Organization of Complex Systems. Pp. 1-27 [2]
- "A system that involves numerous interacting agents whose aggregate behaviors are to be understood. Such aggregate activity is nonlinear, hence it cannot simply be derived from summation of individual components behavior." Jerome Singer in [3]

This thesis has been carried out in the context of a collaboration between the IRD UMMISCO research team and the University of Can Tho (Vietnam), which took the form of a JEAI (*Jeune Equipe Associée à l’IRD*) called DREAM (Decision-support Research for Environmental Applications and Models) since 2012. This work has also been supported by the *“Programme Doctoral International Modélisation des Systèmes Complexes”* (PDI MSC), a collaboration between Pierre-and-Marie-Curie University and the IRD. In this particular context, I have been working closely with other researchers who where developing a number of simulation projects, among which two required the design of complex coupled models. The first one aimed at supporting the design of evacuation policies in case of tsunami in Da Nang city. The second one aimed at designing information systems for supporting the protection against epidemic diseases of crop plants and aquaculture in some of the major Vietnam economic areas. These two examples of complex models, which required coupling multiple submodels, are described in the two following boxes.

Example 1: Da Nang is one of the major port cities in Vietnam (in addition to Ho Chi Minh city and Hai Phong) and the biggest city on the South Central Coast of Vietnam. As pointed out in [4], this city is in the direct passage of the Southeast Asia Sea tsunamis, as shown after the Tohoku earthquake. Therefore, the People's Committee of Da Nang approved, first time in Vietnam, the installation of 10 early - tsunami - warning stations. Supporting this installation, the Vietnam Institute of Geophysics, the National Committee for Search and Rescue, the Da Nang's Steering Committee for Flood and Storm Prevention, and local armed forces and people joined in a tsunami drill focusing on evacuation, rescue and damage repair. More than 6000 people participated in this drill. It was based on a scenario that a magnitude 8.8 earthquake had happened in the west of the Philippines and would cause a 6m tsunami that would strike the coast of Da Nang and encroach the mainland by 700m within two or three hours. It assumed that 6,000 tourists were in the area and 75 vessels with 1,000 fishermen were fishing at the sea.

The drill was successful and the directive board decided to build tsunami scenarios and plans to cope with this kind of disaster.

Several questions could be raised and used as attractive research topic:

- How do the actors understand the risks caused by tsunami?
- What are the most relevant indicators of "community resilience" for these risks?
- What are the existing policies and recommendations for building this resilience?
- What are the conditions for stakeholders to implement these recommendations?
- Are the local communities trained to put these recommendations into practice?

A project combining multi-agent paradigm with data analysis methodology has been established and has reached favorable results ([5], [6], [7], [8], [9], [10]) . In this project, the simulation model assumes the tsunami to be a wave of water that destroys buildings and kills humans and the evacuation model tries to maximize the number of survivals. The evacuation plan itself is based on many factors, i.e., local government policies, rescue resources (such as ambulances, firefighters), the dynamics of city infrastructures in case of disaster. The evacuation model takes into account other models, such as geophysical models, and the government evacuation plan. The set of geophysical models represents the physical elements which can stimulate geo-territory ruptures, resulting in earthquakes or oceanic pulse intrusions which can trigger tsunamis. The government evacuation plan can be completely data-driven, e.g. consisting in a set of procedures established by the local authorities, or be a computational model that simulates the evacuation of people and the management of rescue resources. The coupling of these models allows modelers and decision makers to test and find the best evacuation plan, independently from the variation of type and strength of tsunamis. Thus the core of the case study implementation aims at developing an integrated model that can assess the hazards, vulnerabilities, risks and evacuation solutions in case of a tsunami.

Example 2: In the Mekong Delta region of Vietnam, the provincial agricultural managers are concerned about the regular invasions of Brown Plant Hoppers (BPH), a particularly active rice pest, because of the diseases they carry and transmit to the rice yields. They are also concerned about having an accurate estimation of the distribution of BPH waves in order to support the establishment of different prevention plans. The spread of Brown Plant Hopper waves is affected by many factors. These factors may be environmental factors such as wind speed, humidity, temperature. They could also be the development of urban areas, the change of rice fields and the climate change. BPH invasions represent a complicated, multi-scale spatio-temporal problem with a number of complex dynamics. Firstly, the BPH life cycle is short with three stages: egg, nymph, and adult [11] [12][13]. This insect can reproduce not only on rice but also on some other kinds of grass. Secondly, the BPHs can migrate by following the dominant winds to find their food sources [14][15]. Several studies made by Otuka in 2009 showed that they can migrate on a very large spatial scale [16] [14] There are also other external factors that greatly affect the BPH population such as the farming habits, the use of pesticides, the variability of the weather, fertilizers and seeds used by farmers [17][18]. This situation requires that models aiming at understanding this phenomenon and supporting control policies adopt a multidisciplinary point of view. Modelers in Can Tho University (working on the NOVEL project [19] - Network Optimization Virtual Environment Laboratory) had for instance to reuse and couple at least three existing social, biological and land-use models to be able to apprehend the dynamic interrelations between BPH invasions and control policies.

1.2 Motivations and Research questions

There are nowadays several solutions, like for instance HLA (High Level Architecture [20]), that support various couplings of models but we will see that they do not provide satisfactory answers to the list of issues described in chapter 2. Moreover, the majority of these solutions share a lot of similarities and common practices with modern software engineering technologies but most of the modelers who are facing the necessity to build complex multidisciplinary integrated models are not - and will never become - software engineers. Component-based approaches, model-driven design, or meta-modeling do not signify a lot to the modeling and simulation community, especially in social and ecological sciences. Nevertheless, for the past 20 years, driven by the necessity to build explanatory data-driven models or to answer new types of questions, these same modelers have increasingly used computer modeling formalisms, among them Agent-Based Modeling (ABM, or Individual-Based Modeling). ABM has become ubiquitous in most scientific domains when it comes to explore the interplays between heterogeneous components in real systems. As such, it partially answers one of the challenges raised by the modeling of complex systems by providing a way to build models that are nothing else than a *dynamical composition of sub-models called agents*. But ABM being mostly a conceptual view with no agreed-upon methodological underpinnings and meta-model, it has not, so far, provided any computational solution to the problem of model coupling.

The goal of this thesis is to take advantage of the ubiquity of ABM in modeling and to show that using it for coupling models offers a lot of advantages compared to software engineering approaches, not the least one being that any modeler who knows how to properly design an agent-based model will be able to design integrated models more easily than if he/she has to learn new concepts. I have had to address several challenges to propose a working solution, among which two have revealed quite complex and will be described in details in this manuscript. The first one deals with the conceptual and computational pitfalls of coupling completely heterogeneous models, in particular mathematical and computer-based models. The second one lies in the best solution for aligning the semantics of these different models so that their integration does make sense, which is particularly critical in multidisciplinary models.

This thesis describes the complete working solution, called co-modeling, that I have designed and implemented to address these challenges. I claim that it contributes to and extends the agent-based modeling paradigm, allowing it to represent a complete and coherent approach to multi-modeling and models coupling. Moreover, the conceptual approach that I describe in the next chapters has been fully implemented and tested as an extension to the core of the GAMA agent-based modeling and simulation framework [21]. I have used this implementation to assess its flexibility, composability and reusability in a number of case studies.

The organization of this manuscript is as follows:

- Chapter 2 presents the state of the art of models coupling and analyzes the requirements of the modeling community and some of the drawbacks of current approaches. The analysis focuses on the use of models in multidisciplinary research works such as urbanization, traffic and socio-environmental systems and highlights the most common requirements of modelers in such projects. Then several formalisms and frameworks, already proposed – and used – in models coupling are investigated and their offers compared to the requirements above. A general synthesis of their respective advantages and drawbacks allows to get a clear view of the current choices available for building integrated models.

- Building on the finding that modelers are not software engineers but have nevertheless become familiarized with computer modeling approaches like ABM, Chapter 3 introduces the central proposal of this thesis, an agent-based approach to model coupling, called co-modeling, and its implementation in the GAMA computer modeling and simulation platform. We present its main concepts, its syntax and operational model, and show which responses it is providing to the requirements of flexibility, expressivity, scalability and reusability listed above, allowing modelers familiar with ABM to compose complex coupled models.

- In chapter 4, co-modeling is put into practice on a simple integrated model (Prey Predator toy) which allows us to show its advantages over other approaches, especially in terms of expressivity and flexibility.

- Going beyond the fulfillment of requirements, Chapter 5 shows, on another concrete example, what the interest of using agents for representing models can be, in particular in allowing modelers to build dynamically composed models which can adapt their representation to the systems to model.

- In Chapter 6, finally, we build on the capabilities of co-modeling to demonstrate, this time with the incremental design of a complex integrated model, how completely heterogeneous models can be assembled to answer a question regarding the management of a complex social-environmental system.

- Chapter 7 concludes this thesis by analyzing the outcomes of our research over these past 4 years as well as the perspectives for future works.

Chapter 2

STATE OF THE ART

This chapter presents the definition of coupling in modeling, as well as state of the art of the existing approaches in models coupling. We analyze also the difficulties of coupling heterogeneous models by using existing methodologies and tools and the requirements of the modeling community about multidisciplinary models. A careful overview of existing approaches allows us to analyze their advantages and drawbacks and to propose the initial design of a more flexible solution.

2.1 Why coupling models ?

To answer this question, we present first the definition of model and model coupling and then analyze the advantages of model coupling.

2.1.1 Model

Nowadays, the concept of "model", especially computer model, encompasses several meanings. According to [22], a model is any computer structure intended to represent or simulate some real existing phenomenon (called the "target system"), from which necessary attributes are extracted. According to [23], a model is usually constructed based on a subset of an original prototype's attributes and can be used in place of the original prototype in some situations. The way these attributes or dynamics are chosen depends on the question modelers want to answer: for instance, in ecology, the authors of [24] clearly state that "modeling attempts to capture the essence of a system well enough to address specific questions about the system". The complexity of the target systems is then limited by both the observation and knowledge of researchers about it and the concrete scientific question they want to answer. When this question concerns the interaction of several sub-systems (for example social and biological systems in epidemiology, hydrological and urban models in evacuation planning), it can become necessary to build "heterogeneous models" that reference these different target systems and include multiple components, each of them being a reduced model or system that, sometimes, has been built independently from the others.

2.1.2 Models Coupling

Because the questions asked to models become more and more complex, models coupling has more and more common nowadays, especially in the field of sustainable development, where researchers tend to work in multidisciplinary teams. The requirements of models coupling can come, as above, from the necessity to integrate different models (i.e., urban and climate models, flood and evacuation models, environmental and land use planning models), or to choose the ones which appear to be the most adapted (for instance, to be able to "replace" as will the sub-models composing an integrated model), to understand the same phenomenon at different spatial and temporal scales, or to allow their structure to be incrementally modified to anticipate future changes in the question they are addressing (which is often the case, especially for models used in decision-making processes, where the initial question raised by end-users may evolve over time in the light of new constraints, knowledge or requirements).

In general, models are said to be "coupled" (and not simply, for instance, "juxtaposed") when (1) they can operate independently (i.e. they answer individually a specific question); (2) their integration in a larger model relies on some interactions (exchanges of data, control flow) between them ([25]). Coupling is used when modelers want to analyze heterogeneous systems with multiple levels of details, and when the best models of those systems are an association of existing models. It also occurs when modelers want to answer a question that lies at the interface of different scientific domains, concerns the interaction of different existing components which have been themselves the targets of previous modeling efforts. For instance, questions regarding the impact of climate change on socio-environmental systems are clear examples of the necessity to couple multidisciplinary researches (and hence, models) at different spatial and temporal scales (where the scale at which the climate is changing is not the scale at which adaptation needs to occur).

2.1.3 Advantages of coupling

- **Reusing models allows to cut down the costs and efforts to develop new models.** In the disaster evacuation case study described in Example

1 (section 1.1 of chapter 1), the geophysical model generates data about probabilities of occurrences of tsunami and their consequences (in terms of velocity, wave height, impact on built environment, etc.). This feeds other models among which the one describing the existing government evacuation plans, which, based on this initial feed, selects the best plan available. Then, the evacuation model itself describes the dynamics of people with their individual behaviors and possibly complex interactions (with other people, their environment) when trying to escape the danger and reach shelters as soon as possible. These three models have been individually developed by modelers who are experts in their respective domains, and coupled so as to form a "new" model that answers a different question. Although they share some common properties (water level, buildings, people, etc.), it would have been a complete waste of time to redevelop them entirely just for the purpose of building this integrated model.

In the BPH invasions example (section 1.1 of chapter 1), the spreading of brown plant hoppers in the Mekong Delta is influenced by many factors. These factors may belong to the "environmental sphere" such as the wind speed, humidity or temperature, or the "social sphere", such as the control policies, early-warning possibilities, farming habits, harvesting calendars of farmers or even the "biological sphere", such as the presence of BPH predators, availability of alternate crop species, etc. All these "spheres" are studied by different domains, which have already produced myriads of specific models at different space and time scales [26],[27],[28]. Given the complexity of the whole system formed by the environment, the human settlements and the insect itself, models aiming at assessing and predicting BPH invasions have no other choices than relying on existing models, coupling them in the best possible way so as to build integrated models that will answer new questions (about the best surveillance network, like in NOVEL, or about the optimal mix of control policies, for instance).

- **Coupling models from different domains allows building a more exhaustive, and therefore a more realistic, representation of the real system.** In addition to reducing development costs, coupling models allows modelers to incorporate in their integrated models information that is missing (and needs to be generated by a model) or information that are outside the scope of their competencies. For instance, an expert in evacuation planning will probably not be able to assess the dynamics of a tsunami wave generated by a remote earthquake, but he/she requires realistic inputs to any evacuation planning model he/she might design (otherwise these models may be useless). In that case, he/she will better use another model that generates such an information. The same expert will usually have no practical means to take into account the heterogeneous dynamics of individual people when building evacuation plans (which more or less amount to designating shelters and routes to them), although these dynamics may greatly affect the success of evacuations (appearance of traffic jams, panic movements, etc.). In that case, the only solution is to assess the validity of these plans by submitting them as inputs to individual-based models that are specialized in this area. Improving the realism of models is not only a matter of communication but might prove necessary for them to become reliable and trustworthy, with the hope that this multidisciplinary assemblage of models allows to improve the adaptability of evacuation plans, better manage rescue resources and eventually save lives.

- **Coupled models can be easier to analyze, understand and interpret than complex all-in-one models** Building integrated models as a particular coupling of heterogeneous (sub-)models basically allows to achieve a better separation of concerns, a better independence of parameters. Like in software engineering, this can only improve the readability and documentation of the model, facilitate its maintenance over time, as well as its exploration. In the context of the evacuation case study in section 1.1 of chapter 1, for instance, the sole geophysical model requires more than twenty parameters, and only a tiny fraction of each can be influenced by the outcomes of the other models. Isolating them in a separate model allows to (1) validate this model on its own; (2) provide ways for reusing it in other contexts; (3) understand its influence on other models. Using a coupling approach also paves the way for a more efficient exploration of the parameters of integrated models, as well as a better organization of multidisciplinary work (because it makes clear what intersections, dependencies, or interactions exist between models and therefore between modelers from different domains).

2.2 Different kinds of coupling

A lot of researches have already addressed the problem of coupling multidisciplinary models, but it is in the socio-environmental modeling domain that the richest literature can be found, partly because this domain is intrinsically multidisciplinary, partly because it is a domain where the coexistence of formal (i.e. mathematical, from ecology or environmental sciences) and informal (i.e. computer-based, from social sciences or biology) models is a rule more than an exception. There are many solutions to couple models which are summarized in [29][30], but there are however no unique and agreed-upon classification of the kinds of coupling encountered in integrated models. In [31], coupling kinds are broken down into three categories: coupling based on factors, intermediate coupling and integration coupling. The first category identifies models where the interoperation of the coupled models is entirely described in a common element that orchestrate them (DEVS model (Discrete Event System Specification [32]) or DS model [33]). Intermediate coupling suggests a more decentralized organization, with models adopting interfaces in order to communicate with the others. Integration coupling, finally, represents a more static setting where the coupled models have been modified in order to make a new integrated model.

In [25], the author breaks down coupling techniques depending on a notion of degree (strong, weak) and whether the coupling is technical or methodological. Three kinds of coupling are listed in [34]: final coupling, methodological coupling, technical coupling. Finally, a number of authors also refer to coupling as multi-modeling [35] [36], also knew as a sibling with models coupling,

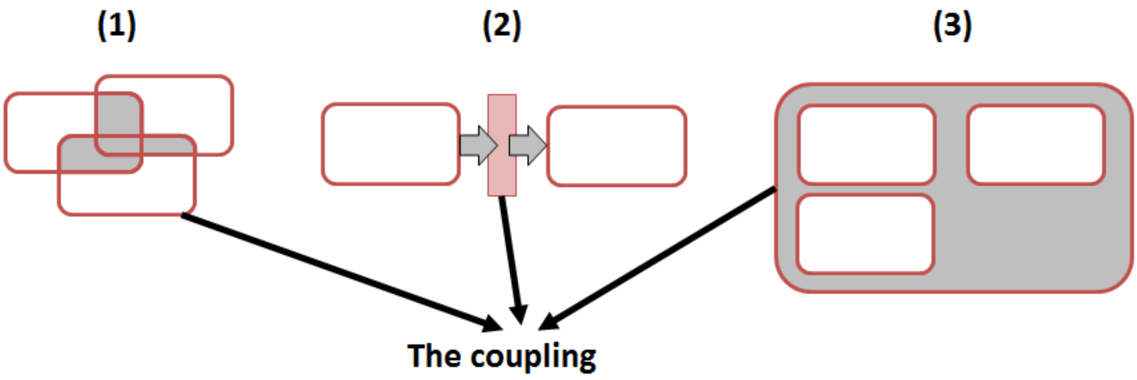

Figure 2.1: The types of coupling models: (1) based on the common factors, (2)Uses intermediate interface and (3) Integrating models. The red-rounded-rectangles are the models.

in particular when the coupled models are expressed in different modeling formalisms.

To make things a little bit clearer, I propose to classify coupling approaches in two extreme groups (between which, of course, a continuum of methods and tools exist): on one hand, the ones that propose what I will call a strong coupling approach between a set of well-identified sub-models, supporting complete and rich interactions between them; on the other hand, a weak coupling approach which mainly relies on data exchanges between multiple models, using interfaces or their equivalent to describe their input and output parameters.

2.2.1 Weak coupling

An interesting example of weak coupling can be found in the work of [37]. In this work, a model called UrbanSim is coupled with another one, called MATSim, to produce an integrated urban mobility model. UrbanSim contains information on residential locations, workplaces and urban development while MATSim provides access to large-scale models of land-use, transportation network and economic dynamics. The two models are solely synchronized through the exchange of data (mobility needs from UrbanSim to MATSim, accessibility indicators in the reverse way). The literature provides other examples. [38] presents an integrated model of a marine environment composed of local ecosystems of pelagic species, each of them being a complete model at its own scale, communicating with the others a number of input and output parameters described as "extrinsic". Another example, presented in [39],couples a community land model with the regional climate model of the West African monsoon. Simple exchanges of data are in general not sufficient in weak coupling approaches because the sub-models, which can be legacy models that cannot be modified, can operate at different scales of space and time and can also have different objectives ([40]). It is then necessary to provide a form of translation (called "coupler" [41]) that takes into account the peculiarities of each model which are often linked to the formalism of the model (e.g., agent-based modeling, discrete event, continuous equations). Numerous works have therefore addressed the problem of combining or coupling models described using different paradigms, like for example [42] on the coupling of hydrodynamic continuous models and individual-based models, [43] on the coupling of physical and social models, [44] on the coupling of continuous and discrete formalisms in ecological models or [45] on the coupling between agent-based models and equation-based models through the use of intermediate graph-based representations.

In the two case studies described in Example 1, 2 (section 1.1 of chapter 1), modelers mainly use a weak coupling approach. In the first case study, it is

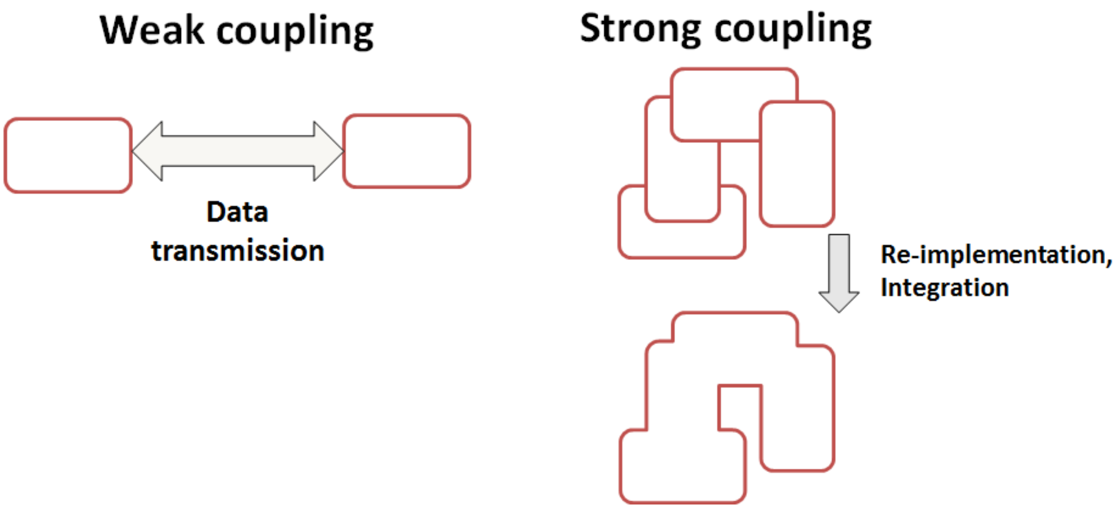

Figure 2.2: Our classification of types of coupling models: (1) Weak coupling and (2)Strong coupling (The red-rounded-rectangles are the models)

due to particular constraints (for example, the geophysical model is built on a particular platform which only accepts input parameters and provides numerical outputs (such as the sea level) but whose components cannot be modified). In the second case study it is more a modeling choice: the number of BPHs, the wind, temperature and humidity values are used for coupling the different sub-models which, otherwise, do not communicate and remain completely independent from each other, facilitating their reuse in other models.

2.2.2 Strong coupling

However, there are situations where the execution or simulation of sub-models need to be more controlled, and where even complex exchanges of data or parameters between models is not sufficient. This can be the case, for instance, if different models of the same phenomenon are integrated together and if only one should be chosen at a given time or for a given spatial scale. It can also be due to the necessity of running some models, for example stochastic models, repeatedly in order to improve the confidence interval of their outputs. In those cases, a stronger coupling, which involves some functional controls in addition to the exchange of data, must be used ([46]). Strong coupling usually relies on operational architectures or frameworks that can provide a way to express the control over sub-models. They can either use an existing modeling paradigm (like [47] which proposes an "agent-centered" approach in which different modeling formalisms can be translated to individual or agent-based models) or make use of specialized software architectures dedicated to the functional coupling of models, like the High-Level Architecture (HLA) ([48], [20], [49], [50], [51]), the Discrete Event Systems (DEVS) ([32], [52], [53]) or the Functional Mock-up Interface ([54], [55]).

Both groups have their advantages and drawbacks: on one hand, strong coupling can produce more integrated solutions but may lack flexibility, preventing, for instance, sub-models to be easily replaced; on the other hand, a pure weak coupling approach, while more flexible in theory, requires the design of an interface that can limit the types of sub-models that can be used.

In this thesis, we defend an intermediate approach between these two extreme views, which relies on a dynamical combination of weak and strong coupling in order to provide more flexibility and genericity to modelers. To show the effectiveness of our approach, we will list in the next section the challenges of models coupling we intend to address (and how existing approaches succeed or fail in addressing them).

2.3 Challenges of model coupling

Models coupling is not completely different from software engineering in terms of challenges. In the world of Software Engineering, many measures of the quality of a software have been proposed. In the book [56], the assessment of a good architectural design includes measures of reliability, performance, security, maintainability, flexibility, testability, portability, reusability and interoperability. Some of them are clearly outside the scope of this thesis (e.g. security), dependent on how the underlying software is implemented (e.g. portability, performance), completely covered by Software Engineering alone (e.g. testability, interoperability) or dependent on each other (e.g. reliability and maintainability require some sort of flexibility). Modeling also brings its own constraints, like the existence of time and space scales, for instance. Given all that, we propose that coupling models approaches should, a minima, satisfy the four requirements of *reusability, scalability, expressivity,* and *flexibility*. *Reusability* represents the ability provided by the approach to reuse existing models, even if they are expressed in different formalisms. *Scalability* represents the ability to cope with models that rely on different spatial and temporal scales. *Expressivity* represents the capability to fully and explicitly describe the coupling of models: how they interact together, how they exchange information and how they are controlled. Finally, *flexibility* represents the ability to dynamically add, remove or replace any of the sub-models coupled together.

2.3.1 Reusability

Reusability in software engineering has two sides: one is purely technical (e.g. reusing libraries in different languages, making sure a software produced with a given language can be linked with other pieces of software) while the other is both conceptual and linked with an idea of "controlled genericity" (e.g. a piece of software can be reused because it has been designed in such a way). Models coupling, which often has to deal with the necessity to reuse existing models, faces the same problems and there is probably NOTHING new we can add to the existing solutions provided by Software Engineering. It however adds a third, more semantic, constraint, which is linked to the inner formalism in which the models are expressed. As a matter of fact, it is less and less unusual to encounter integrated models in which several formalisms are used to express models. For instance, [57] uses a combination of three sub-models: one based on the differential equations paradigm, one based on some variety of cellular automata and an agent-based one. All three formalisms rely on completely different, almost antagonistic, meta-models and the translation between their inner representations is not something trivial, especially if they are tightly coupled. Reusability in models coupling, as opposed to its meaning in Software Engineering, is then more concerned with the possibility of expressing, in an integrated model, easily and transparently, how sub-models can be integrated and translated than with the technical possibility to do it.

2.3.2 Scalability

When coupling different models, modelers often face a problem of scale translation, either because the existing models produce outcomes at different scales of time and space, or because they require input parameters or data at specific scales. Equation-based models, for instance, often operate at aggregate scales (population, whole territory, continuous time scales, etc.), while using agent-based models imply some sort of locality (individuals, parcels, discrete time scales). It is not always the case, of course, but when it is, it requires some possibility to describe the levels at which each model is operating and how data and control can flow between these levels. For instance, the spatial scales of two of the components of the evacuation model (the model of tsunami and that of the evacuees) are completely different.

Moreover, the same model can be used at different scales and must then be fed with the right parameters and interpreted accordingly. For example, in the BPH invasions model, light-traps (which are static traps used to capture and count insects) are present and modeled with their zones of influence at the scale of the town/district level but also at larger scales (quarter, province, region). Some sort of spatial discretization and aggregation operations are necessary to make the same model work in these two contexts.

The necessity to translate data between the temporal scales of the different sub-models is also crucial. For instance, the data of the meteorological stations, used in the BPH invasions model, are produced monthly, while the trap density data are produced daily (see MARD [58] – Meteorological Data by Month, Years, and Stations, Ministry of Agriculture and Rural Development). The integration of these two models requires that some translation occurs at some point, and this translation cannot be completely independent from the models themselves (i.e. it might be possible in some models to simply average the meteorological data over one month, but it cannot in some where other models depend on the occurrence of specific weather events, like rainfalls).

A good, *scalable*, coupling mechanism then needs to offer some ways to express explicitly the translations between the spatial and temporal scales of each sub-model, even if they are complex – and even if they themselves become an inherent part of the integrated model itself, like another sub-model (e.g., if we have monthly meteorological data with an average rainfall, when should we decide to make rainfall events occur on a daily or weekly basis ?).

2.3.3 Expressivity

The two requirements above imply some sort of *multi-modeling*: in our view, an integrated model should be seen, if we want it to be *reusable* and *scalable*, as both a collection of sub-models **and** a model of the connections between them. This implies, in a way or another, that the coupling infrastructure uses a *modeling language* to describe the coupling, like an agent-based model uses an agent-based language to describe the interactions between agents or a mathematical model uses equations to describe relationships between variables. And it is of course better if this modeling language is already a language that modelers can apprehend easily because it leverages their experience in designing and writing models. Unfortunately, most of the existing solutions [59] clearly separate the languages in which models might be expressed and the language (when they have one) that is used to describe their coupling. Most of them require modelers to learn completely new languages – sometimes new concepts – in order to describe even simple couplings. For example, HLA and FMI are two powerful frameworks for multi-simulation, but they require the use of structures, concepts and languages for describing couplings that are completely outside the world of modeling (and which are never used to describe models).

2.3.4 Flexibility

Finally, besides the previous aspects that mostly concern modeling, an important requirement of a good coupling approach is the level of **flexibility** it offers, not only at design time, but also when exploring, experimenting or testing models, i.e. during simulations. By flexibility, here, I designate the possibility to dynamically change the structure of the integrated model at runtime, for example swapping a sub-model for another, allowing to add or remove models, or even changing the coupling as well. This possibility can become a necessity, for instance in integrated models where some sort of learning mechanisms is implemented or where several sub-models can play the same role (with different accuracies, different requirements, etc.). For instance, some required data can be produced, for a given period, from a historical dataset and, for a period where this information is missing, by a model.

2.4 Existing solutions in modeling/simulation

Most of the existing integrated models found in the literature make use of "ad-hoc" coupling techniques that are designed for the sole purpose of providing an infrastructure for one specific set of sub-models and are almost impossible to be reused in other contexts. For example, in [37], the connection between UrbanSim et MATSim is completely bound up with this particular instance of integration, and there is no way one can reuse it, adapt it or extend it, for instance with an environmental model. It is the case when models come from different domains (urbanization and transport [37], environment and pelagic resources [38], terrestrial ecosystem and regional climate change [39]), but also when they come from the same domain [40]. Data exchanges are also often "ad-hoc, even if some effort of genericity has been made in some integrated models, like the one presented in [41], which uses an explicit coupling mechanism, called "coupler", that exchanges input and output data between models operating at different spatial and temporal scales.

Finally, the same conclusion can be reached when models that use different modeling approaches are coupled: the coupling of an hydrodynamic model and an individual-based model in [42], the coupling of a multi-agent model and a GIS model in [46] or [44], the coupling of a physical model and a social model in [43], all use "ad-hoc" approaches that cannot be easily generalized to other modeling contexts or problems.

This problem is not new and represents one of the reasons why a part of the modeling community has spent some efforts in designing architectures or frameworks with the purpose of supporting complex models coupling: the High-Level Architecture (HLA) [48], the Discrete Event Systems (DEVS) [32] or the Functional Mockup Interface [55]) are three robust examples of this trend, which we will present and analyze in the following sections in order to understand if they meet the requirements expressed above.

- **High level architecture** HLA, defined initially in the military sector, is mostly used in human training to perform tasks and analyze scenarios in a simulated world. HLA integrates mechanisms for synchronizing heterogeneous simulators when they exchange data. The principle of HLA is to consider that simulators are assemblies of a Federation. An interface RTI (Runtime Infrastructure) assures the synchronization of exchanges between the Federations. HLA is defined by three core elements: the template object model (including HLA Federation Object Model and HLA Simulation Object Model), the interface specification with Runtime Infrastructure, and HLA rules. Despite the numerous advantages in the handling of completely heterogeneous simulators and having a quite complete implementation, the main problem of HLA is its complexity for non-computer scientists, which makes it out of the reach of most modelers.

- **Discrete Event System Specification** DEVS was initially a formalism proposed to model discrete event systems. Its interest for model coupling lies in its recursive definition: a model described in DEVS (with its set of inputs, outputs, states, etc.) can be considered as either "atomic" or "coupled", in this case it is described with additional features like the models it couples (which can be themselves atomic or coupled), a translation function and the influences between these models. An 'atomic' model is considered as a sub-model with contributing parameters (its set of input, output events, sequential states; its time advance, its external/internal transition function and its output function). A 'coupled' model is specified by three other parameters: the set of atomic models it couples, the translation function between the inputs and outputs of these models, and the influences between models (which use specific 'ports' to control them). DEVS represents a particularly elegant way of describing the coupling between models and, although it seems to be well adapted for building composite models, the fact that it relies on a formal and deterministic internal description of models prevents it to be really useful when having to assemble stochastic or complex legacy models.

- **Functional Mockup Interface** FMI is an industrial standard for co-simulation, where each sub-model or simulator is wrapped and exposed to the others using a functional interface which specifies how it can be accessed or manipulated. FMI is an independent approach for model exchange that supports 'black box' model exchanges (i.e. there is no need to know how a model works if its interface is sufficiently well described). This standard has been developed to satisfy the requirements of standardization, availability, ease-of-use, adoption, accompanying documentation and maturity of models coupling in the industry. As such, it is not really known and used in the academic world, although it offers some interesting features, in particular in terms of standardization. FMI also suffers from its software engineering origin, imposing a whole new way of describing models that does not rely on any modeling paradigm used in environmental or social sciences.

Despite being operational (and being used elsewhere), HLA, DEVS, and FMI are almost completely absent from the literature on hybrid/integrated modeling ([60]) for addressing environmental, socio-environmental, urban or ecological issues. The probable main reason is that they mostly address the software engineering side of the problem of models coupling and require modelers to learn and master quite complex techniques. In the worst cases, sub-models even need to be completely rewritten or redesigned to adapt to the interface required by these coupling infrastructures, which is something industrials might be willing to do, but which would represent too much energy and efforts for academic modelers. Furthermore, these techniques use formalisms and languages that are different from the ones commonly used to build socio-environmental models (like agent-based models, Cellular Automata, or mathematical models).

Beside these three standard solutions, several other contributions have been originating from within the modeling community, especially from the designers of simulation platforms, which offer their own coupling methods and their own definition of spatial, temporal and data exchanges, however usually tied to their platform. The advantages and disadvantages of the main solution are summarized in the following table:

	Advantages	**Drawbacks**
DEVS	- It is a good conceptual approach for describing the assemblage of possibly interacting atomic sub-models within a larger coupled model	- It is mostly a conceptual approach, implemented in a few tools.

	- It offers a very good support for designing complex discrete-event models and simulations.	- It is difficult to describe models that do not follow discrete-event dynamics (continuous time, discrete time-based models). - It does not really offer supports for complex behavioral or environmental declarations. - It is very limited when it comes to couple stochastic models.
HLA	- It is easy to reuse and combine simulations in an interoperable manner. - Distributed simulations are well controlled with data communicating and synchronization of actions.	- It is a pure technical solution in term of a standard that does not support dynamical changes in the coupling structure. - It does not focus on the modeling of the coupling, which is disseminated in the various structures. - It mostly requires to re-implement legacy models in the C++ language or to build quite complex wrappers around them.

FMI[61]	- It supports the coupling of physical models emanating from different domains - It allows exporting and importing model components in industrial simulation tools (FMI for Model Exchange) - It standardizes co-simulation interfaces in nonlinear system dynamics (FMI for Co-Simulation) - It provided the most efficient co-simulation interfaces between electronic, mechanical and software models. The primary goal of FMI is to simulate and analyze these models.	- Co-simulation interfaces are only available (and documented) for the engineering and industrial domains - Its support in terms of modeling is quite limited, i.e. it mainly focuses on the co-simulation and exchanges of models, not really on the description of their coupling.

MADKIT[62]	- It supports the definition of quite complex and dynamical organizational structures between software components called agents, similar in practice to the agents used in ABM - It uses a recursive definition similar to DEVS but less limited	- It is not specifically dedicated to modeling (in spite of the presence of a modeling language called TurtleKit, similar to the one found in NetLogo) - It does not offer any support in terms of multi-formalism (besides the ability to write agents in Java, Python or C++)
The Model Coupling Toolkit (MCT)[63],[64]	- It is an effective method for coupling many parallel models to form one high-performance coupled modeling system.	- It is dedicated to atmosphere and ocean general circulation models, land-surface models, and dynamical sea-ice models. - It uses a set of outdated Fortran90 modules defining derived data types and routines
VLE [65]	- It is an interactive modeling GUI that provides an easy access to the DEVS framework.	- It does not have a specific language dedicated to modeling.

		- It does not offer a lot of support for multi-formalism models or various data sources. - It suffers from the same problems than DEVS regarding stochastic models.
Netlogo[66]	- It is the most popular modeling platform - Its model library is plentiful and covers most of the research domains.	- It allows modelers to develop only medium and small models, mostly for academic purposes. - It does not support the coupling of models, even simple ones.
AToM3[67]	- It is a powerful code-generating tool for multi-formalism modeling. - Instead of building the whole application from scratch, it only requires to specify, in a graphical manner, the types of models that need to be coupled.	- It only works with models that can be transformed into graphs.

ARCHON[68]	- Coupling is supported by agents which encapsulate existing softwares as components of agents.	- It is not adapted, due to performance problems, to large models. - It is not really maintained anymore.
OSIRIS[25]	- It is flexible and generic. - It allows to couple different physical processes. - Its GUI helps to define the characteristics of coupled models.	- The sub-models cannot be modified at runtime - It does not support explicit spatial coupling.
JADE library[69]	- It is a high-performance software platform for delivering solutions targeted to complex business problems.	- It does not have a specific language dedicated to modeling. - It does not offer a good support for the description of spatial processes.

2.5 Conclusions

Although the approaches presented in this chapter support, each with its own constraints, some degree of interaction between sub-models in integrated models and some degree of expressivity in the description of their coupling, they nevertheless have the following drawbacks:

- Many of them, besides the three “standards” that are HLA, DEVS and FMI, are not generic and do not support the definition of coupling architectures that can be easily reused in different domains or contexts.
- Most of them, besides HLA and DEVS, do not support the definition of explicit spatial and temporal scales and the translation between them.
- When they are not bound to a particular modeling paradigm (like DEVS), which cannot support all the varieties of models (like stochastic models), they impose to use languages and concepts that are not familiar to the modelers.
- Finally, the majority of them (besides DEVS and possibly HLA) have never been tested against the design of complex data-driven, multi-formalism integrated models.

However, some offer interesting concepts and implementations: the encapsulation of legacy models as *agents* provided by ARCHON or OSIRIS, the recursive organization of atomic and coupled models found at the heart of DEVS and MADKIT, the flexibility of the ‘temporal controller’ provided by HLA are such examples of structures that have been designed by necessity to cope with models coupling and that should be reused (or adapted) in any new proposal. After having analyzed (and tested) most of them, I propose a new approach for the problem of models coupling in the next chapter, that both builds on the existing works and expands them so that they can become easily usable by modelers. This approach is conceptually close to the one presented as "multi-model ecologies", described by [70] and comes with a complete agent-based implementation in the GAMA platform [21]. Beyond the different integrated models we present in this manuscript, this proposal aims at being as generic as possible and constitutes the main outcome of my research.

Chapter 3

CO-MODEL: AN INFRASTRUCTURE FOR SUPPORTING THE DYNAMICAL COUPLING OF HETEROGENEOUS MODELS

The previous chapter has pointed out the drawbacks of current approaches to models coupling. Building on the finding that modelers are not software engineers but have nevertheless become familiarized with computer modeling approaches like ABM, this chapter introduces the central proposal of this thesis, an agent-based approach to models coupling, called co-modeling, and its implementation in the GAMA computer modeling and simulation platform. We present its main concepts, its syntax and operational model, and show how it can provide an answer to the requirements of flexibility, expressivity, scalability and reusability listed above, allowing

modelers familiar with ABM to compose complex coupled models.

3.1 Introduction

The chapter 2 has listed some of the requirements of modelers regarding architectures or solutions for coupling models, which are reusability, scalability, expressivity and flexibility. A software framework, based on a conceptual methodology satisfying these four requirements, is the central contribution of this thesis. The method provides a language to describe any specific coupling of models, allows to encapsulate most of the existing models and supports the translation of multiple scales of space and time, all of which using simple extensions to an existing modeling language. The result can be executed and simulated just like a normal model, offering features such as dynamical addition, removal and on-the-fly replacement of sub-models.

3.2 Basic concepts

The method I propose sports the name of "co-modeling" (a contraction of "coupled modeling") and considers coupled models (called "co-models") as a particular sort of agent-based models, where agents wrap one or several instances of the models to be coupled, with their life-cycle, operations, collaborations and conflict resolution mechanisms, which can be inspired by the numerous works already published on multi-agent systems ([71],[72],[73],[74]).

3.2.1 Co-model

In this perspective, a co-model is nothing more than an agent-based model in which some of the agents represent other models, allowing a recursive description similar to that of MADKIT or DEVS, and in which the coupling between its "micro-model" agents are described exactly the way interactions or collaborations between agents can be described in an ABM. This approach leverages the expressivity of ABM and allows to reuse its neutrality in terms of agent implementation to support multiple model formalisms (viewed as particular behavioral architectures). The only requirement this approach imposes on its micro-models is that they explicitly describe their spatial and temporal scales so that they can be accessed and modified by other micro-models or the co-model they are belonging to.

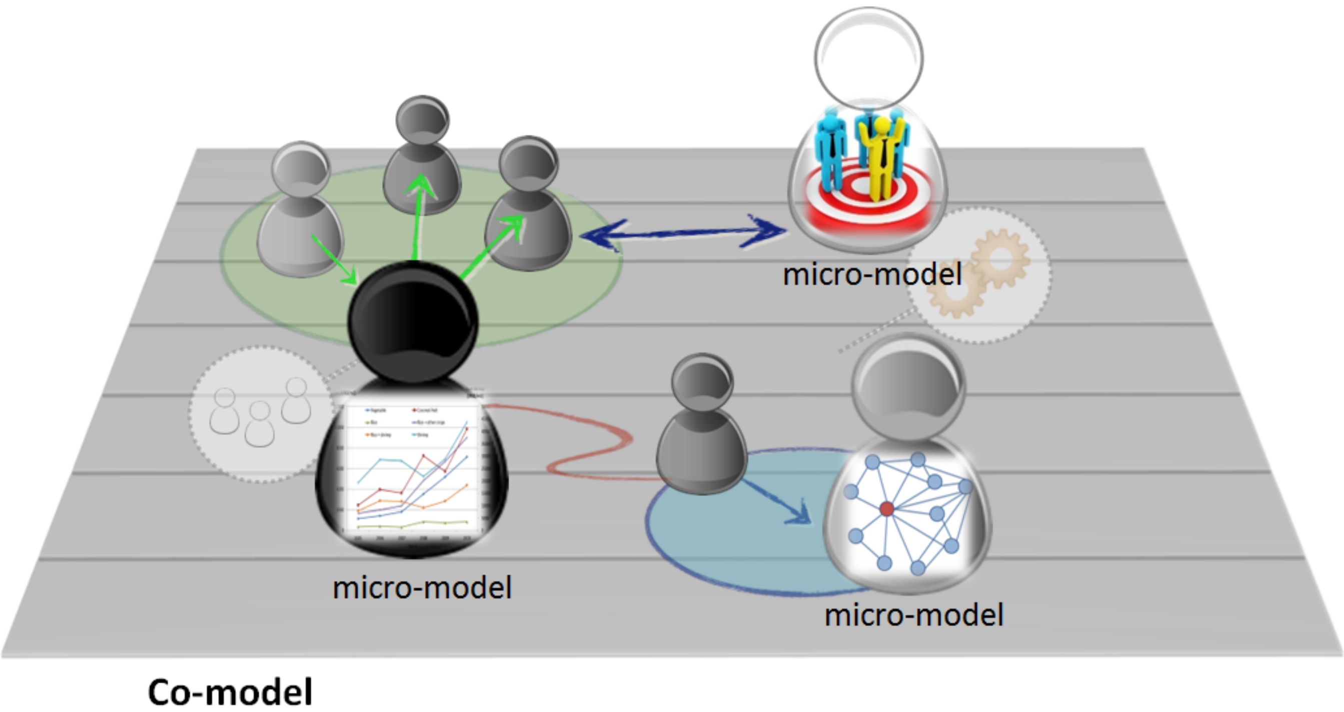

Figure 3.1: Co-models extend the base concepts of agent-based modeling formalism by allowing agents to be models themselves

Figure 3.1 presents a simplified view of the proposal. A co-model is made up of a collection of agents, among which some represent other models (called micro-models). There are not, conceptually, any differences between 'regular' agents and 'micro-models' ones.

3.2.2 Micro-model

A micro-model is a model that has been instantiated, as an agent, in a co-model. This agent is considered as both a 'wrapper' (in that it gives access to its attributes, behaviors, etc.) and a 'representative' of this model in the co-model (as a matter of fact, in our implementation, the agent is literally an instance of this model). In order to complete the loop, micro-models can of course be co-models too (figure 3.2). This recursive organization allows modelers to declare as many times as needed the structure of "models as agents".

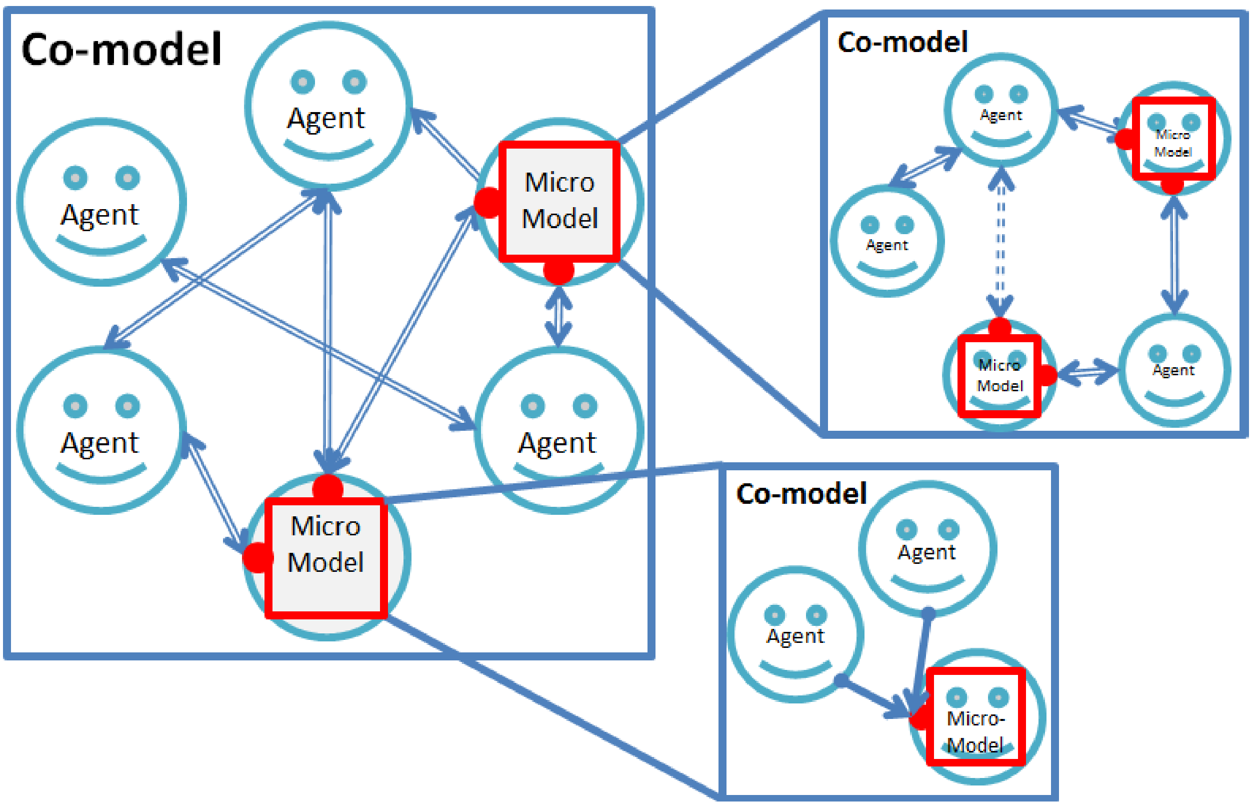

Figure 3.2: Micro-model can be co-models too

3.3 Integration in the GAMA platform

3.3.1 Why GAMA?

A proposal about models coupling needs to rely, somehow, on at least one implementation in an existing programming or modeling environment, effectively used by modelers, otherwise it is bound to remain a pure conceptual exercise. In his survey [75], Kravari compares 24 modeling platforms using different criteria: usability, operating ability, pragmatics, security management and application domains. According to the latter, which is an important criterion for modelers and an indication that it can be usable for building interdisciplinary projects, GAMA is today the platform used in the widest variety of domains (internal numbers, obtained from the developers, estimate its user base at around 2000 to 3000 users). It extensively supports data-driven models, creating agents from almost any data sources, including spatial databases, and supports the simulation of large models (up to millions of agents with complex behaviors).

GAMA is built around a dedicated high-level agent-oriented language (GAML), which sports a clean and simple syntax that makes it accessible for modelers who may have difficulties in programming in conventional languages like Java or C++. GAMA is also built as an extensible open-source platform that allows modelers to develop new features and functions, in Java, to satisfy their needs if they are not yet available. All these characteristics made it the ideal candidate for implementing and testing a model coupling architecture.

3.3.2 Implementation

My approach mainly consisted in enriching the GAML language in two directions: (1) by allowing a model to reference other models as micro-models (and not as regular imported models); (2) by allowing these references to be instantiated as agents. Although the result seems like a small addition to the language, it has required a lot of under-the-hood efforts, which are detailed below.

I firstly had to alter the existing meta-model of GAMA (depicted in Figure 3.3 [76], [77]), which was a "traditional" agent-based meta-model supplied with a less traditional multi-level extension allowing to declare agents as compositions of other agents. The first addition, conducted together with my PhD supervisor (one of the main designers of the platform), consisted in clarifying the status of models and simulations in GAMA, by making the meta-class of models inherit that of regular agents. In that way, models acquired a status equivalent to that of *species* in GAML (and similar to *classes* in object-oriented approaches), and their instances, called *simulations*, became regular agents. The second modification consisted in clarifying the role of *experiments* in the meta-model: initially, experiments were designed to behave like "main classes" in Java, a set of specifications for building instances of models, with different parameters, outputs, etc. I reinforced this role by making experiments the mandatory access points to imported models (so as to preserve their encapsulation) and, since experiments are also agents, the only possible representative of a micro-model in a co-model. Declaring and instantiating a micro-model then consists in creating an instance of one of its available experiments (which in turn can give access to the instance of model it is encapsulating).

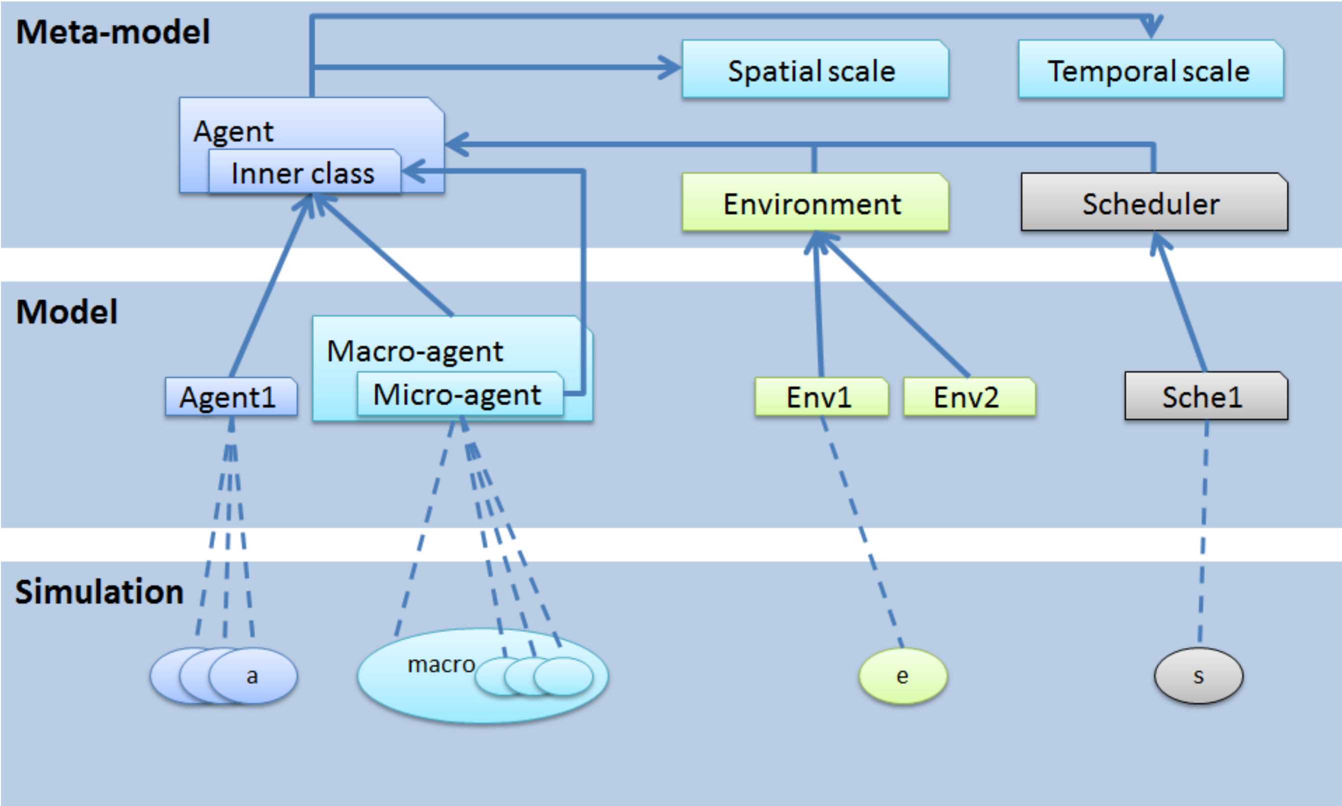

Figure 3.3: The meta-model of GAMA [76]

These modifications (3.4) enabled me to fully implement the fact that a co-model in GAMA is represented as a regular model in GAML that wraps several micro-models. This wrapping uses a special form of importation, where imported models are provided with a model identifier (which enables, later on, to have access to the experiments defined in them). Any model can thus contain the definition of several micro-models representing different couplings that will be instantiated and possibly executed by its instances (the simulations). These changes in the general meta-model of GAMA were of course not possible to handle with a simple mechanism of extensions or plugins, and had to be introduced for the whole platform. Since GAMA has reached version 1.7, they are incorporated in the standard meta-model and available for all modelers (see https://github.com/gama-platform/gama/wiki/Introduction).

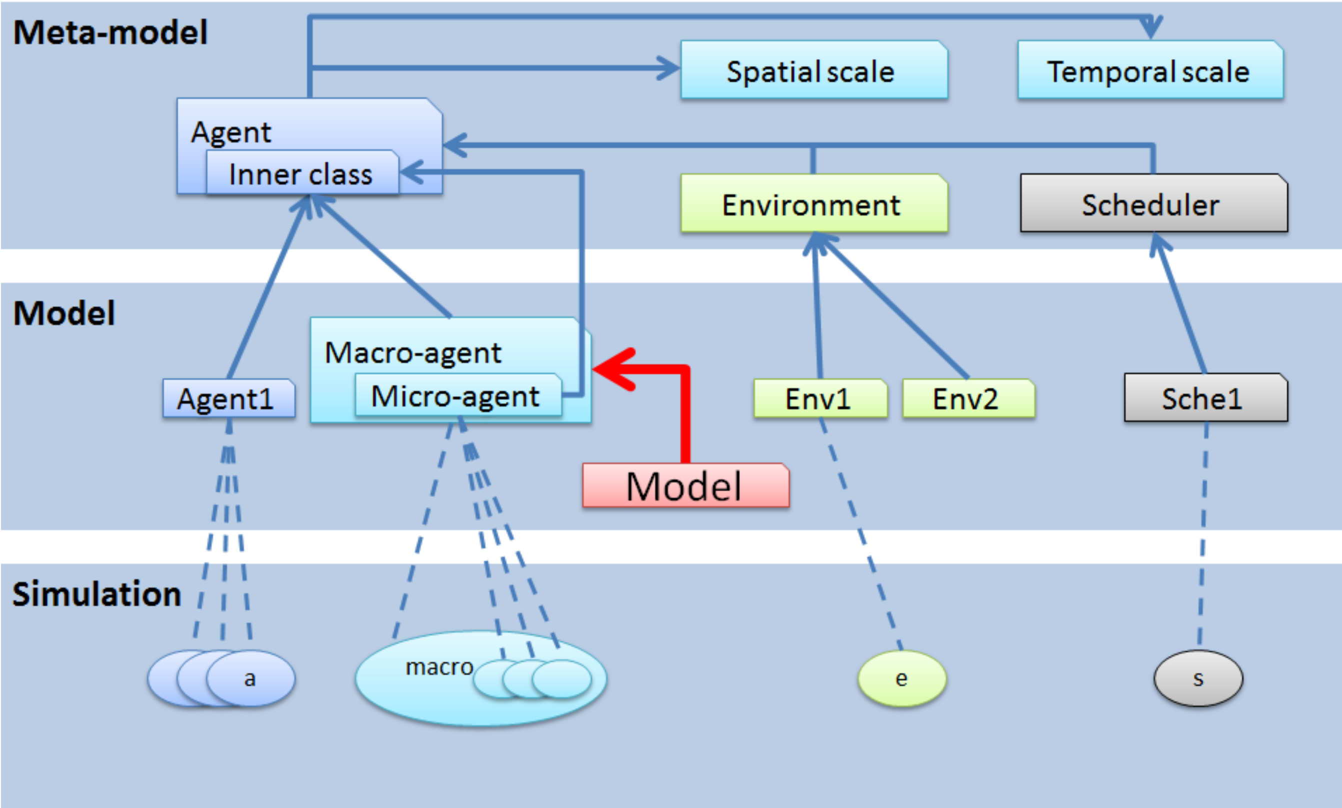

Figure 3.4: Extend a simplified meta-model version of current agent-based modeling platforms to support coupling models (changes are in the dotted rectangle)

3.3.3 Portability

The co-modeling approach proposed here is easy to generalize *conceptually*, as it relies on the really simple assumption that: "any model can be considered as an agent in an agent-based model". *Operationally*, things are less trivial, as it requires the presence, in the target platform or framework, of a multi-level or recursive organization of agents where agents can be composed of other agents. It also requires that spatial and temporal specifications of models be explicitly described and accessible from other agents (as it is the case in GAMA, where each agent, including of course models, can describe its inner environment and inner scheduling process). The portability of the co-modeling concepts in other computing environments is then strongly dependent on the availability of these

```
import "marketProduct.gaml" as myMarket
import "SeaLevel.gaml" as mySea
import "environmentChange.gaml" as myEnv
```

Figure 3.5: Importing model as micro-species

features in these environments. It should be possible to do it in Madkit, for instance, but not in NetLogo.

3.4 Example of use (syntaxes)

This section presents the different steps necessary for the construction of an integrated model based on the co-modeling approach.

3.4.1 Importation

First, we need to identify and import the micro-models that need to be coupled. Importation requires to define an identifier for each of them. This identifier is used as an alias name, within the co-model, for the imported micro-model. Figure 3.5 presents an example of such an importation, where the "import" statement is followed by the path to an existing model and its identifier.

3.4.2 Instantiation

Once micro-models have been imported, they need to be instantiated like any other agents to be scheduled in the simulation. Instantiation in GAML uses the keyword “create”, followed by the species of the agent to instantiate and possibly other attributes, like the number of agents to instantiate, the initialization of their attributes, etc. However, in GAML, a model is simply a description of a system; the operational part of the model, used to build simulations, is described in its experiment(s). Since micro-models are intended to be "executed" in the simulation of the co-model, the modeler needs to specify which of their declared experiments should be instantiated. Figure 3.6 shows an example of syntax for the instantiation of the three imported micro-models with the default parameter values defined in the experiment chosen (in this example, "market-grow", "SaltIntrusion" and "EnvChange" are supposed to be valid experiments described in, respectively, "myMarket", "mySea", and "myEnv").

```
init {
        create myMarket.marketgrow;
        create mySea.SaltIntrusion;
        create myEnv.envChange;
```

Figure 3.6: Instantiate agents of micro-species

3.4.3 Execution

Thanks to the multi-level modeling in GAMA [77] and our coupling approach, the execution of a micro-model is carried out by asking the micro-model agent to do the simulation step by step (see Figure 3.7). Besides, modelers can access all the behaviors and attributes of the micro-models and exchange data with and between them. We will see in the next chapters more examples of these possibilities of exchange, but it is important to note right now that the spatial and temporal scales of micro-models can be accessed naturally using three of their accessors, respectively "shape", which defines its geographical boundaries, and "step" and "starting_date", which both define its default temporal attributes.

```
        ask myMarket.marketgrow
        {
            do _step_;
            myself.price_rice <-  priceRice;
            myself.price_shrimp <- priceShrimp;
            myself.price_coconut <- priceCoco;
        }


        ask mySea.SaltIntrusion
        {
            loop times: 10
            {
                do _step_;
            }
            myself.envOut <- outParcel;
        }
```

Figure 3.7: Ask agents of micro-species to simulate and exchange data between models

3.5 End of chapter

This chapter has presented, albeit in a concise way, the principal contribution of this thesis: an extension of the ABM meta-model that promotes model as agents and allows them to be included in other models. This extended meta-model has been used to replace the one used in the GAMA modeling and simulation platform, allowing, after a few modifications to the GAML language, to instantiate models within models and execute them as if they were agents. This implementation is fully recursive, as agents can be representative of models which are, themselves, co-models (i.e. composed of other models). The control and data exchanges between micro-models is handled in a natural way by way of accessors to their attributes and execution of their actions (including the ubiquitous *_step_* action, present on all models). With this implementation, building a complex integrated model is not different than building a (complex) agent-based model. Moreover, since GAMA handles multiple formalisms gracefully (either as extensions to the GAML language or by defining plugins that wrap models or data sources written in other languages – native executables, scriptable environments like R or NetLogo), this addition can be used as a good starting point for multi-modeling efforts.

Chapter 4

Demonstration and usage

This chapter presents the implementation of a simple integrated "Prey-Predator" model using the proposed co-modeling approach. Based on this practical example, we analyze also the advantages of co-modeling compared with the state of the art approaches, especially in terms of expressivity and reusability.

4.1 Objective

This chapter presents the implementation, in the proposed co-modeling approach, of a practical example (a toy model with no other purposes than to serve as a demonstration of its usage) in order to show how we can address the challenges of models coupling presented in chapter 2. The capabilities of the proposed approach, especially the interoperability of its infrastructure, are evaluated using the following criteria:

- Reusability: Modelers could immediately reuse existing models for building the toy model without too much modifications and efforts.

- Expressivity: The coupling of models could be declared and described explicitly, without any *black box* effect
- Scalability: Models operating by default at different spatial and temporal scales could be coupled together.
- Flexibility: Micro-models could be dynamically added, removed or swapped during the simulation of the co-model.

4.2 Toy model description

The example I use in this chapter is a classical toy model (called "Prey-Predator co-model" in the following). This model attempts to describe the dynamics of two populations part of a simplified food chain: a population of "preys" composed of simple individuals wandering around and looking for "resources" in their environment, and a population of "predators" that feed on the preys. The reproductive success of each population is dependent, as it is classical in this family of models, on the ability of its individuals to find enough food to survive. This model relates to a wide literature, in computer science, economy ([78]), ecology ([79]), or sociology ([80]), etc. but the goal of this particular implementation is neither to be realistic nor to bring new findings, just to serve as a demonstration of the co-modeling approach.

4.3 Toy model implementation

I take the hypothesis, in this example, that we are in a fairly classical situation: on one hand, there exists a model that describes the dynamics of a population of animals, situated in an environment where they can feed on some resources; plenty of similar models exist in the literature, but we chose to implement a deliberately simplified one for the sake of the demonstration; on the other hand, we have another model, completely independent from the first one (it could have been designed by other modelers), that describes two populations of animals: preys, which are supposed to be “available" and their predators. Both models can be experimented, studied, calibrated with respect of real data, and used for their own purposes (i.e. depicting the dynamics of their respective populations). The goal of the exercise is, then, to build a *third model* with a slightly different purpose: understanding the combined dynamics of two populations, one of preys and one of predators, each of them feeding either on environmental resources (preys) or preys (predators), like a short and simplified food-chain.

4.3.1 The animal-resource model

The first model, which I will call “animal_resources", is then composed of a population of preys that move around in a shared environment and hunt for its resources. Preys are supposed to follow some random walk, avoid other agents (whichever they are) and consume resources when they find some. Each move costs them some energy (we suppose they have some), each consumption of resource adds to it. When the energy is below a certain threshold, they die; above a certain threshold, they give birth to a new prey with which they share equally their energy.

Such a model is very easy to implement in GAMA. Since any model defines, by default, a “world", we give this world two behaviors: at initialization, to create animals and, at every step of simulation (in a so-called reflex), to create some resources. By default, all these created agents are placed randomly in the world. Animals belong to a specific species (the equivalent of a class in GAML) called animal, which is decorated with a skill called moving. This provides its instances with default displacement behaviors (like wander, goto, follow, etc) that can be parametrized if needed (with a speed, a target, etc.). We also provide animals with three perception functions: resource_perceived (if a resource can be seen in the world), resource_available (if one can be eaten), and animal_around (if any agent, including other preys, is present). Each of these functions is tested at every step of the simulation and leads to a possible behavior: moving towards a resource, eating a resource (which takes the form of "killing it and adding some energy") and fleeing randomly to avoid others. Resources are also represented by agents, but they don't do or know anything (hence their "empty" definition). Note, at the end of the definition, the "experiment Simple;" statement. Experiments are, in GAML, the way to instantiate and run the simulations of a model. They are used to setup their parameters and define their outputs. Multiple experiments can be defined for a given model.

```
model animal_resource
global{
   init {
      create prey number: 100;
   }

   reflex {
      create resource number: rnd(10);
   }
}
species resource {}
species animal skills: [moving] {
   int perception <- 10;
   int predation <- 1;
   int energy <- 100 update: energy -1;
   bool resource_perceived -> { !empty(resource at_distance: perception)}
   bool resource_available -> { !empty(resource at_distance: predation)}
   bool animal_around -> { !empty((agents - resource) at_distance:
      perception) }
```

```

   reflex when: resource_available {
      ask (resource closest_to self) {
         do die;
         myself.energy <- myself.energy + 1;
      }
   }

   reflex when: resource_perceived {
      do goto target:(resource closest_to self);
   }

   reflex when: animal_around {
      do wander;
   }

   reflex when: energy < 0 {
      do die;
   }
}

experiment Simple;
```

4.3.2 The prey-predator model

The second model represents similar dynamics, but this time between two “mobile" populations, respectively called *predator* and *prey*. Both move randomly and predators are opportunistic: whenever they are close to a prey, they eat it (they could of course have more sophisticated behaviors). In order for predators to know which agents they can still “eat", this model maintains a list of available preys (every time one is eaten, it is removed from this list by its predator). And it creates, from time to time, some preys (since these ones do not reproduce). Nothing too extraordinary, then, in this model, except that it is of course possible to define displays (like the one on Figure 4.1) in order to follow what is happening.

```
model predator_prey
global{

   init {
      create prey number: 1000 returns: created_preys;
      available_preys <<+ created_preys;
      create predator number: 100;
```

```
    }

    list<agent> available_preys <- [];
    reflex {
        create prey number: 10;
    }

}
species prey skills: [moving] {
    reflex {
        do wander;
    }

}

species predator skills: [moving] {
    int energy <- 100;
    int predation <- 1;
    bool prey_available -> { !empty(available_preys at_distance: predation)}

    reflex {
        if (prey_available) {
            ask available_preys closest_to self {
                do die;
                available_preys -< self;
                myself.energy <- myself.energy + 1;
            }
        do wander;
    }

}

experiment Simple;
```

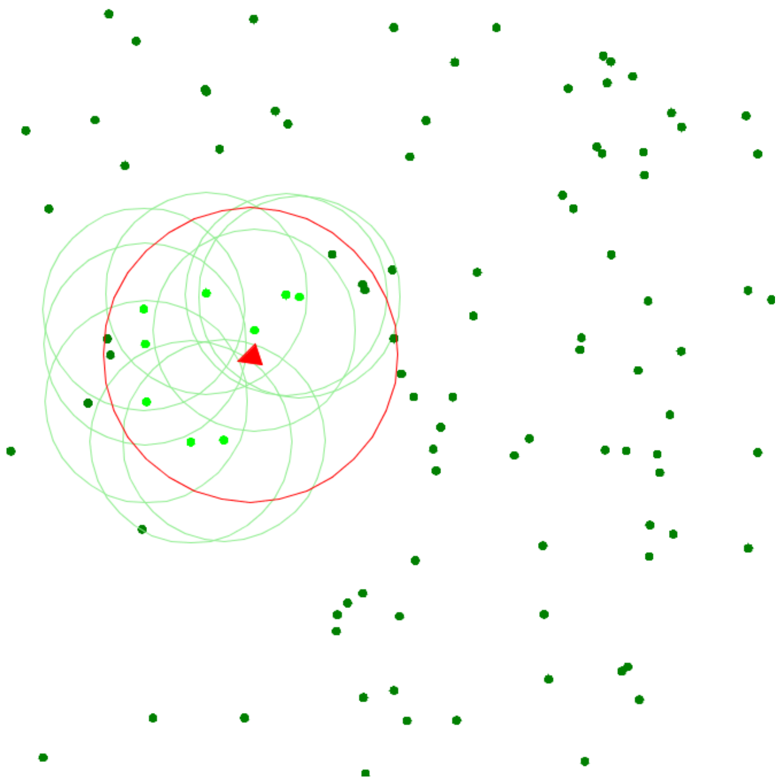

Figure 4.1: The display of the Prey Predator simulation

4.4 Co-modeling

Once we have declared the two models above, assembling them in one coupled model (a co-model) to study the population dynamics of the whole food-chain is fairly easy and does not require any more knowledge than the one used for building regular models.

4.4.1 Step by step co-modeling the prey-predator co-model

Step 1

The co-model is declared as any other model, with the exception that it imports now the two previous ones as *micro-models* and provides them with a unique identifier.

```
model coupled

import "animal_resource.gaml" as Preys
import "predator_prey.gaml" as Predators
```

Step 2

Like we instantiated animals or predators in the previous models, we can instantiate models. The only difference is that a model can only be instantiated through one of its experiments (in our case, both micro-models have only one, called “simple"). This step will create one instance of each model, which will now be components of the co-model’s simulation and, most importantly, inhabit its “world" (with its own spatial and temporal scales, which will then become the spatial and temporal scales of the two micro-models).

```
    create Preys.simple;
    create Predators.simple;
```

Step 3

Micro-models can behave in their new world like any agent. To ask them to execute one step of simulation, we just need, literally... to ask them. Their attributes, actions become accessible to the co-model.

```
ask [Preys.simple, Predators.simple] { do _step_;}
```

Step 4

However, asking the two micro-models to execute their steps will have them run in parallel, like two different models, which is not exactly what we intended to do. So the last step required for building a co-model consists in defining the “glue" between the micro-models, i.e. the coupling mechanisms: what interactions and exchanges of data are necessary to couple the models? In our current example, the coupling mechanism is simple: we need to substitute the preys of the second micro-model by the animals of the first one to create a food-chain linking predators-preys/animals-resources. This sentence translates into the GAML code below, within the reflex. Each step of a simulation of a co-model consists in: (1) emptying the predator-prey of its prey agents (in case some are still alive); (2) tell this micro-model that its available_preys is now the population of animals of the animal_resource micro-model; (3) ask the two micro-models to execute one step of simulation.

This mechanism is defined below as a behavior of the world of the co-model. But it could perfectly be defined in another species of agents belonging to this co-model and in charge of scheduling micro-models. We can then see how easy it would be to define different strategies of coupling *within the same co-model*, each of them handled by a different agent.

```
model coupled

import "animal_resource.gaml" as Preys
import "predator_prey.gaml" as Predators
global{
   init {
      create Preys.simple;
      create Predators.simple;
   }

   reflex coupling_method {
      ask (Predators.simple) { ask simulation.prey {do die;} }

      Predators.simple.simulation.available_preys
         <-list(Preys.simple.simulation.animal);

      ask [Preys.simple, Predators.simple] { do _step_;}
   }
}
```

Listing 4.1: full co-model representing the co-model prey-predator

4.5 Discussion

This chapter has presented a first, deliberately simple example of the usage of the co-modeling approach in GAMA. We discuss below how this implementation satisfies the requirements listed in Chapter 2.

4.5.1 Reusability

The coupled model presented in this chapter is a good example of how legacy models (the animal_resource one and the predator_prey one) can be imported, altered if necessary and reused quite easily (using a few lines of code) in a new model. Moreover, this same coupled model can then be reused itself in another coupled model, for example to represent a complete food chain where the predators of the first micro-model become the preys of the second one. For instance:

```
model second_coupling
import "coupled.gaml" as First
import "predator_prey.gaml" as Second
global{
   init {
      create First.Simple;
      create Second.Simple;
   }
   reflex {
      ask (Second.simple) { ask simulation.prey {do die;} }
      Second.simple.simulation.available_preys <-
         list(First.Simple.simulation.prey);
      ask [First.simple, Second.simple] { do _step_;}
   }
}
```

As it is, this model looks a lot like the first coupled model. It reuses the same pattern of composition and paves the way for a more generic approach (i.e. to model food chains of arbitrary length by reusing over and over the same micro-model with different parameters. In this example, we do not change the default parameters of the coupled models for the sake of readability. However, it is of course possible (for instance, at instantiation time, to provide a name for each species, different speeds, perception radiuses, etc.). The only constraint for reusing models is that they provide enough “accessors", i.e. sensors and actuators regarding their internal state. In this example, the coupling would not have been so simple without the presence of the “available_preys" accessor (which can be accessed and modified from outside) in the “predator_prey" model. Without it, that is if the predators of the model were only using their own population of preys as a potential reservoir, it would have been possible but more complicated algorithmic notations would have been necessary to couple the two models (i.e. the coupled model would have been in charge of maintaining the equivalence between the animal population of the first model and the prey population of the second one). However, a better solution is to rely on the recursivity offered by co-models and use so-called *wrapper models*. Since micro-models can themselves be co-models, it is possible to declare intermediate micro-models that would act as wrappers of the micro-models and provide explicit accessors to their internal state(s), simplifying a lot the writing of the coupled model. For instance, we could define such a wrapper model for the predator_prey micro-model:

```
model predator_prey_wrapper
import "predator_prey.gaml" // the model is imported "normally"
experiment Simple {
   action replacePreys(list<agent> new_preys) {
      ask simulation {
         ask prey {do die;}
         available_preys <- new_preys;
      }
   }
}
```

and the coupled model would simply have to import the wrapper and call this action every step without worrying about the exact name of the species in the imported predator_prey model or what it means to “replace" the preys.

4.5.2 Expressivity

Unlike other solutions that support the coupling of heterogeneous models, the co-modeling approach allows to describe the interactions between models *in an explicit way and using the same exact modeling language as the one it is implemented in*. Co-models being only available for the GAMA platform as of today, it requires of course to write the coupled models and the wrappers around the micro-models in GAML. But as new implementations will be developed, this necessity will progressively vanish. And the nice aspect today is that, for a modeler fluent in GAML, there is no need to learn a new language or new concepts: manipulating models, wrapping them, coupling them is exactly the

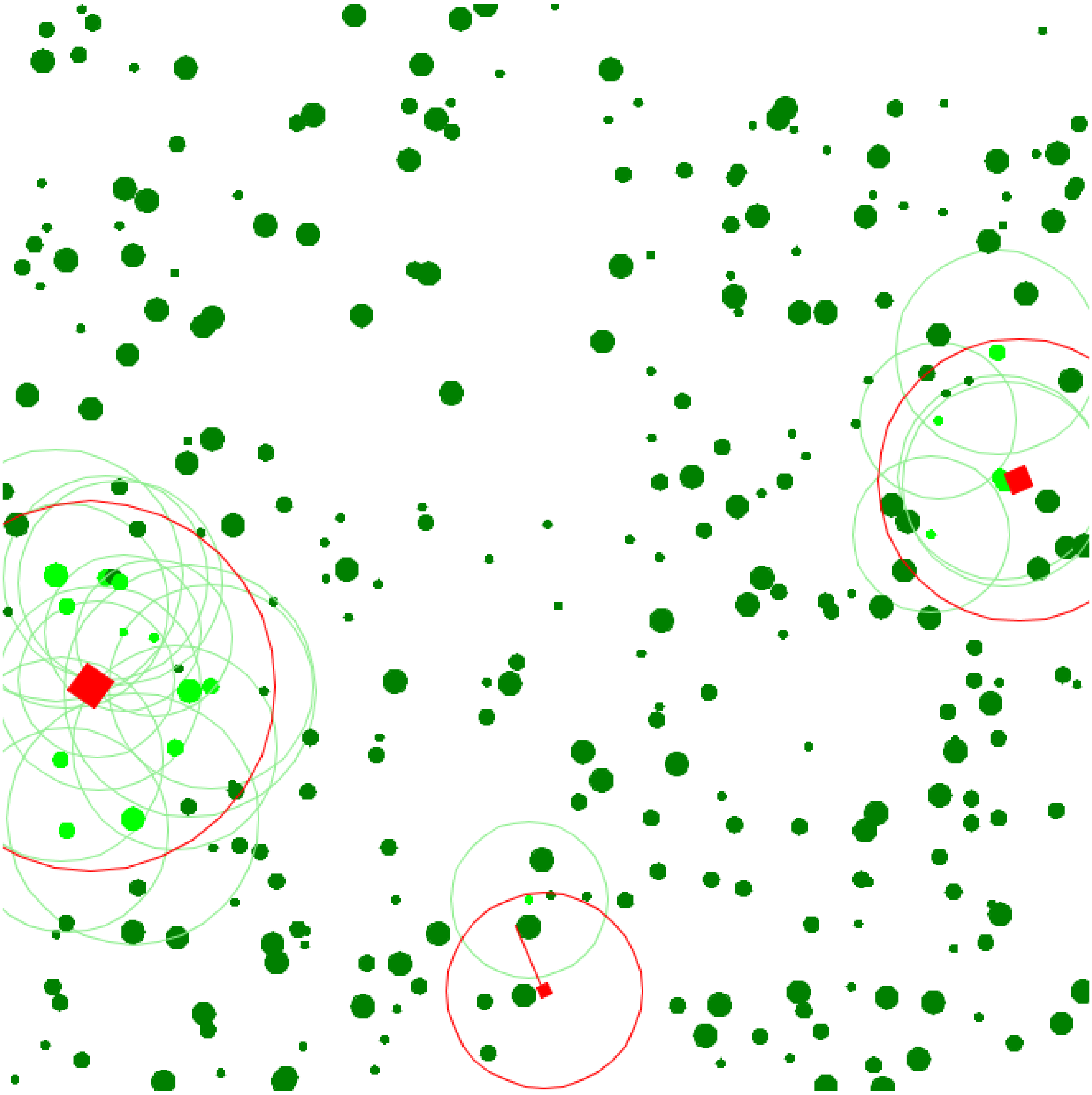

Figure 4.2: Display of the food chain co-model. The green circles represent the preys and the red squares represents the predators. The food chain among three types of preys and three types of predators is modeled. The difference between different types of species are represented by their shape sizes.

same as working with agents.

4.5.3 Scalability

The possibility to alter and adapt the spatial and temporal scales of each model has not been explicitly used in this example, although it is implicitly realized by just coupling the two models. It is in any case quite easy to implement by manipulating two accessors present in every model in GAML: shape and step. Shape allows to redefine the spatial boundaries of a model, step its temporal resolution (i.e. the duration of each time step). The two can then be changed at initialization time in each micro-model to have them adapt to new boundaries and time resolution. Note that, by default, the boundaries and time resolution of the co-model become the ones of each micro-model is nothing is specified (all the agents are “merged” into the same environment).

4.5.4 Flexibility

In the toy example presented in this chapter, the flexibility of co-modeling has not been really demonstrated, in particular the ability to add, remove or substitute micro-models at runtime. This will be the subject of the next section.

In conclusion, this chapter has presented, via a deliberately simple example, how the conceptual contribution of this thesis, once implemented in a modeling and simulation platform, can concretely be used. With a few lines of code, any model can be imported and transformed into an agent within another model. This opens a lot of interesting perspectives for building integrated models.

Chapter 5

Dynamic choice of the best representation of a phenomenon

Going beyond the example presented in Chapter 4, this chapter shows, using another concrete example, the interests of using agents for representing models. In particular, that it allows modelers to build dynamically composed models which can adapt their representation to the systems modeled. The interest of using co-modeling in that case lies in the ability to choose the most appropriate representation of a phenomenon given the dynamic context of a model. For example, in epidemiology, it is sometimes more appropriate to use an equation-based model (if the modeled population is large) than an agent-based model (if the modeled population is small). The reason is that equation-based models, being aggregate models, can be quite accurate at large scales, where the individual behaviors are averaged, but can fail to reproduce small scale dynamics when only few individuals are present and where the behaviors of these individuals matter a lot and cannot be averaged. Building a coupled model that, based on some conditions,

switches between the two representations is a classic example of the interest of models coupling. I show in this chapter how it can be accomplished thanks to the co-modeling approach and what this approach exactly provides to the modeler. In particular, I show that, although it does not provide any "magical" solution to the semantic problem of how to align the two models (i.e. how to switch from a continuous to a discrete representation and conversely), it nevertheless provides the support for modelers to describe the solution to this problem in an explicit way, as a new model, so that it can be fully documented, modified, and reused.

5.1 Objective

This chapter shows the capability of the co-modeling approach to support the modeler in describing, in a model, how a same phenomenon (the transmission of a disease in a population) should be represented depending on the size of the population considered. The model in question will be called, throughout this chapter, the "switch model". Initially, we describe it as a normal model that contains a mixture of ABM and EBM formalisms that both implement the SIR (Susceptible - Infected - Recovered) [81][82] epidemiological model at the scale of individuals and at the scale of their population. This model is then transformed into a coupled model, and we describe the advantages resulting from this transformation.

5.2 Modeling context

Agent Based Modeling and Equation Based Modeling are two possible modeling paradigms of dynamical systems like epidemiological systems. Equation Based Models (EBMs) usually describe dynamical processes at an aggregate scale (at the population level in ecology) while Agent Based Models (ABMs) describe the same processes at a more local scale (at the individual level in ecology). Each approach has advantages and drawbacks, which mainly depend on the question, but also on the data available and the scale at which the processes need to be represented. These conditions determine the way the model is constructed. For global processes, for which EBMs are usually more appropriate (if the question, of course, does not imply some sort of individual variabilities), the model is constructed with a small number of parameters and without any individual variability, assuming that mean field approximations (or other aggregate functions) conveniently describe the dynamics at the global level. ABMs are relevant when the individual variability is believed to have a strong effect on the dynamics emerging at the global level. Additionally, they allow explicit representations of the interaction network of individuals when its topology is known to influence the dynamics of the system and the emergence of properties at the global level. ABMs also offer the possibility to easily integrate spatial and social network information.

Apart from these conceptual aspects, the community to which the modeler is belonging has a strong influence on the choice of approach. A strong knowledge in mathematics is needed to understand and build equations. As a counterpart, mathematics offer powerful tools to analyse EBMs, provide a lot of in-depth information about the dynamics such as the equilibria and the long term dynamics. The ABM paradigm is more intuitive and ABM platforms such as Netlogo or GAMA propose modeling tools aiming at a wider audience. Given their respective properties, some researchers have worked on the comparison or the coupling of both approaches. An interesting example of coupling between EBMs and ABMs can be found in a particle transport model [83] where the output of an EBM is used in an ABM to describe the dispersal of fish larva. The transport model is based on an oceanic current model which is itself based on physics and Partial Differential Equation. However, to my knowledge, there are very few models of strong coupling of an ABM with an EBM where both models really interact, i.e. use the outputs of the other. Such a model has been developed by [84] regarding pedestrian movements. The model is based on an ABM describing the movements of individuals in the streets of a city. Each road segment between two crossroads can be replaced by a mathematical transport model in order to reduce the amount of resources needed for the simulations. At each intersection, the ABM feeds the EBM with the number of individuals entering the road segment, then the EBM generates agents at its end. Another similar model is proposed in [85], [86]. However, in these different cases, the coupling is completely inflexible and ad-hoc, leaving no real possibility to either reuse the models in other contexts or reuse their "coupling patterns" easily. The model presented in this chapter aims at demonstrating that the co-modeling approach can provide a flexible solution to this problem, without sacrificing the accuracy of the model, and without requiring the modelers to fundamentally change their habits. This model also inspired with the adaptive coupled models which can be composed several micro-models being aware of its other appropriated representations [87].

5.3 Definition of micro-models

The first step has consisted in implementing in GAMA a hybrid model, named “Switch”, that tightly combines the two paradigms, equations and agents (figure 5.1). In this model, people are represented by agents when the density is low and by equations when the density is higher. A simple tilting mechanism for switching between the two representations has been implemented as well.

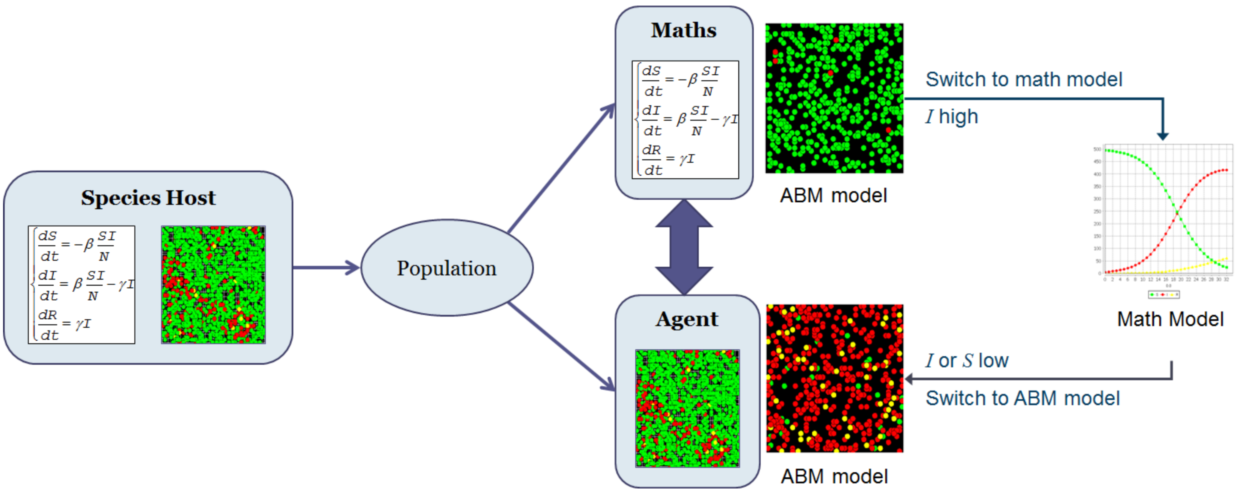

Figure 5.1: Representation of the dynamics of the "Switch" model

Both models are based on the same assumptions. They involve two processes: contamination and recovery. The ABM model adds spatial interactions and dispersal. The mathematical model is indeed a mean field approximation of the ABM and represents the dynamics at the global scale, while the ABM shows the dynamics at the local scale. The contamination and recovery processes occur frequently with a "uniform distribution" over time.

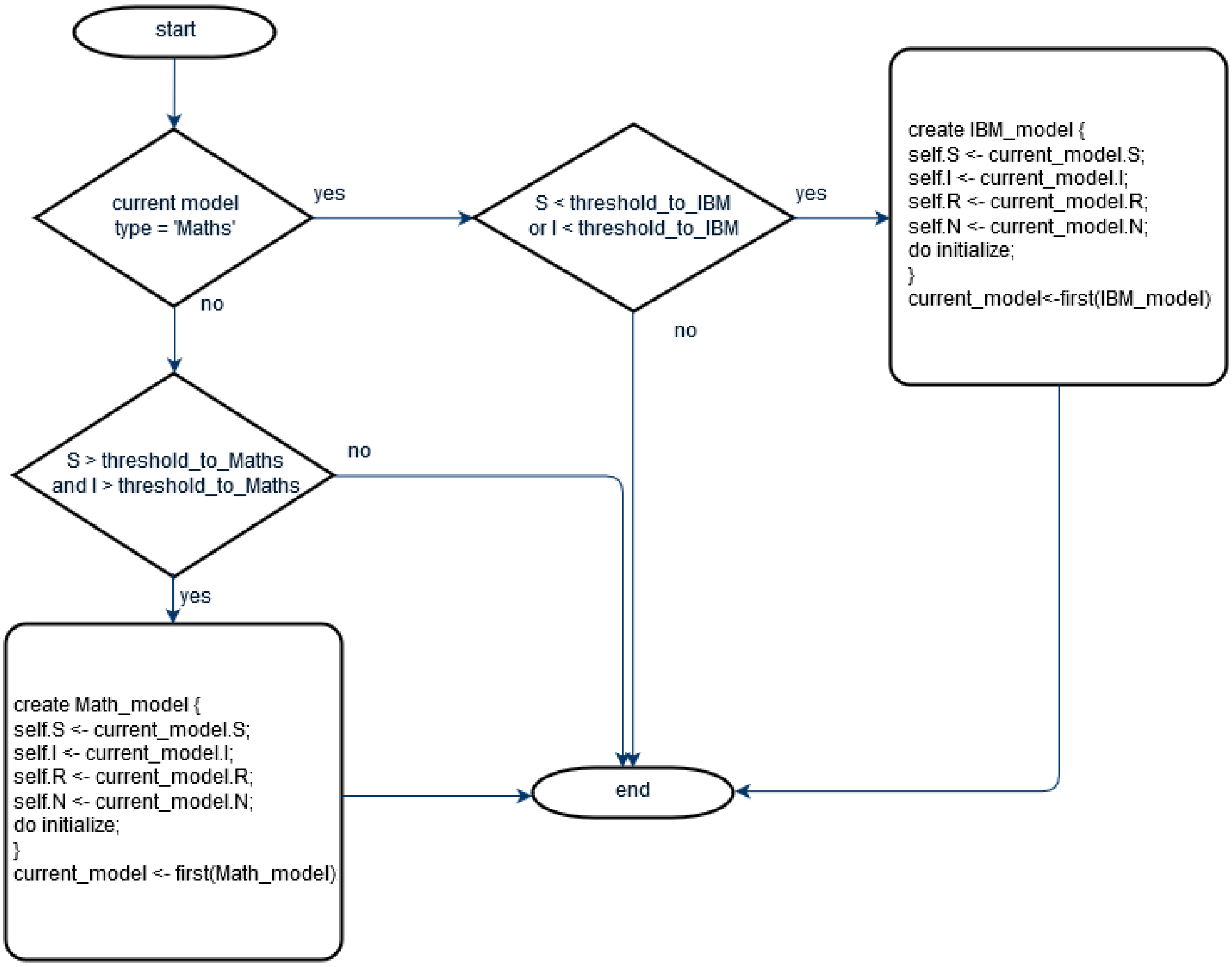

Figure 5.2: The controller in the switch model which changes the current model to ABM or EBM based on the population condition.

The two models are based on the theoretical SIR model assumptions. Individuals can be in three different states. They can be susceptible (S), i.e. disease-free, potentially contaminable by contact with an infected individual (I). After some time, infected individuals recover from the disease (or die). They are assumed to be in a recovered state (R) which means that they are immune to the disease and do not take part anymore in the infection dynamics. The models involve the following processes:

- infection: transmission of the disease from infected individuals. This depends on the contact rate between susceptible individuals and infected individuals
- recovery: infected individuals heal and recover from infection.
- movement:individuals are assumed to move within their environment. There are two types of movements, one random and one not, (figure 5.3).

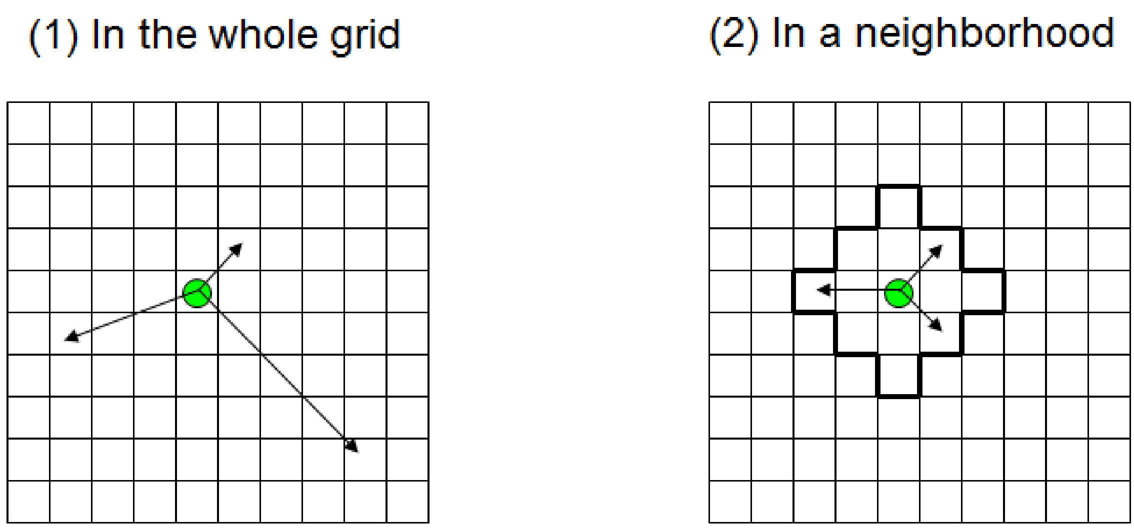

Figure 5.3: Two types of displacements of agents in an environment.

Hypothesis found in both models:

- Recovery rate: the remission rate is very similar in the agent-based model and the equation-based model. In the ABM, the parameter gamma is the probability to recover per time units. In the EBM, the parameter gamma is a mean field approximation, which means that the number of recovered individuals given by the EBM is exactly the expected number of recovered individuals given by the ABM and there is no infection occurring at the same time. The stochasticity of recovery rate appears with low I populations, otherwise both models fit.

- Contact rate: in the present models, contacts are defined in a similar way for the mathematical model and the agent-based model. In the agent-based model, two individuals are considered to be "in contact" if they are in each other's vicinity during a time step. In the mathematical model, space is not explicitly represented, but the average number of neighbors can be determined. The stochasticity of contact rate appears according to the size of neighborhood (strong variability in the number of neighbors) and the speed of people (a low speed means no mixing, the neighborhood proportion of R and I may greatly vary).

The general model could be found in the List 5.1 in which the species SIR_model is the parent of the two species of agents respectively named ABM_model and Math_model.

```
species SIR_model schedules: [] { float S; float I; float R; int N;
   string model_type <- 'none';
   action initialize ;
   action remove_model {do die;}
}
species IBM_model schedules: [] parent: SIR_model {
   string model_type <- 'IBM';
   action initialize {
    create Host number: S {
```

```
        is_susceptible <- true;
        is_infected <- false;
        is_immune <- false;
        color <- #green;
     }
     create Host number: I {
        is_susceptible <- false;
        is_infected <- true;
        is_immune <- false;
        color <- #red;
     }
     create Host number: R {
        is_susceptible <- false;
        is_infected <- false;
        is_immune <- true;
        color <- #yellow;
     }
    }
    action remove_model {
     ask Host {
        do die;
     }
     do die;
    }
}
species Math_model schedules: [] parent: SIR_model {
    string model_type <- 'Maths';
    float t;
    equation SIR {
     diff(S, t) = (-beta_maths * S * I / N);
     diff(I, t) = (beta_maths * S * I / N) - (delta * I);
     diff(R, t) = (delta * I);
    }
    reflex solving {solve SIR method: "rk4" step: 0.01 ;}
}
```

Listing 5.1: The general view of the Switch model

The control flow in Figure 5.2 is translated in GAML as:

```
species switch_model schedules: [] {
    int threshold_to_IBM <- 45;
    int threshold_to_Maths <- 50;
    bool start_with_IBM function: { (initial_S < threshold_to_IBM or
       initial_I < threshold_to_IBM) };
    reflex switch_to_IBM when: (current_model.model_type = 'Maths') {
     if ( current_model.S < threshold_to_IBM
        or  current_model.I < threshold_to_IBM) {
        create IBM_model {
           do initialize;
        }
        ask current_model {
           do remove_model;
        }
        current_model <- first(IBM_model);
     }
    }
    reflex switch_to_Maths when: (current_model.model_type = 'IBM') {
     if ( current_model.S > threshold_to_Maths
        and  current_model.I > threshold_to_Maths) {
        create Math_model {
           do initialize;
        }
```

```
        ask current_model {
            do remove_model;
        }
        current_model <- first(Math_model);
    }
  }
}
```

Listing 5.2: The controller in the Switch model which changes the current model to ABM or EBM based on the population condition

It is to be noted that this model relies on the possibility, in GAMA, to use a differential equations system to describe the dynamics of agents and to integrate (using different integrators) this system at each simulation step. This is an enhancement of the GAML language I have developed myself early on in my PhD thesis and which is now used regularly by GAMA users to write equation-based models directly in the platform [21]

5.4 Transformation of the Switch model into a co-model

The Switch model uses a tilting mechanism based on the number of susceptible and infected individuals. When the density of the population is high, it switches to the EBM and switches back to the ABM when the density is low. This mechanism is fundamentally a mechanism of "coupling", with the drawback that the micro-models considered are directly implemented in the Switch model and cannot easily be replaced (for instance other SIR-based representations, involving different hypotheses, different datasets, or other behaviors for individuals). A transformation of this model into a co-model has then been handled in order to (1) increase its flexibility; (2) clearly separate the concerns between the representation of the disease (SIR models) and the coupling mechanism; (3) offer the possibility to reuse this coupling mechanism for other models sporting different representations.

5.4.1 Dynamics of co-model

In the co-model resulting from the transformation, all the concepts, work flows and algorithms presented above are kept intact. Just a small change needs to be done regarding the declaration of the micro-models. A micro-model is not anymore a species that inherits from a SIR_model, but rather a whole model (for example a legacy model belonging to a set of library models) which is accessible through an interface named xxx_wrapper. The interface is only committed to provide accessors, such as get_num_s, get_num_i, get_num_r, and set_num_s_i_r. As long as it can be imported and manipulated by the wrapper, the legacy model can be a black-box model using whatever formalism or paradigm the modeler needs. The co-model just needs to ask, through this interface, the legacy model for the current values of S, I and R and, once it has computed the size of the population using an appropriate model, to return this number to the imported model.

The co-model is then only concerned with the definition of the best switching mechanism (for example to fit a particular dataset) between the two micro-models imported (which can, of course, be more numerous, if other representations are available, or run repeatedly with their outputs average for building confidence intervals in the case of an ABM representation). The switching mechanism I have implemented is represented by the behavior of a Switch agent that belongs solely to the co-model and whose role is to control the data transfer between the co-model and the legacy model(s). The number of susceptible, infected and recovered individuals are attributes of this agent. Its pseudo-code is as follows:

```
if(condition of switching){
current model <- get the relevant model
current model . set the the input to the legacy model
ask current model { do step }
sir < - current model . get the output from legacy model
}
```

It is easy to see that this switching agent, implemented only for regulating the execution of SIR models, could be further generalized and abstracted so as to provide a basis for families of "switching models", where the condition for switching can be different (i.e. not only the size of the population, but maybe their localization too, or some other attributes) and the wrapper interface different as well.

5.5 Conclusion

Thanks to the wrapper models introduced in the transformation process, the Switch co-model, contrary to its predecessor, can reuse any model that accepts the three parameters S, I, and R and returns them. Here, I have only reused legacy GAMA models that were immediately available, but it is of course possible to wrap any other model (for example, models built in R or Matlab) as long as the necessary plugins are available for GAMA. This "design pattern" (abstraction of the inner mechanisms of models through an interface) is nothing new in the world of software engineering, especially that of object-oriented approaches. But, in the modeling community, it constitutes a small revolution, by allowing to clearly separate the concerns between the possible representations of a phenomenon and the choice of this representation given a particular (for example, data-driven) context. The separation can even go further, since the “micro-models” that are coupled can perfectly wrap something else than a model, for example a data source. In which case the switching co-model presented here could be reused in another co-model and provide, for some temporal conditions, either the exact number of S, I and R cases present in a dataset or an evaluation of these cases given by a model for the missing years. It is also easy to see how this construction can be easily reused for more advanced interactions between models: each step of the switching agent could for instance perfectly consist in running two or more SIR models, average their outputs and returning this average as the result, so as to compensate the biases introduced by their representation. This resulting co-model is in any case a demonstration, on a practical example, of the possibilities offered by the co-modeling approach for building flexible integrated models. I will show in the next chapter how precious they are in the context a more ambitious and complex model.

Chapter 6

Incremental design of a complex integrated model

This chapter presents how the co-modeling approach can be used in a context of incremental design of a complex integrated model. The model in question is part of a work undertaken in common with [57] to build a land-use change model able to reproduce correctly the land-use change dynamics of a specific region of the Vietnamese Mekong Delta and improve actual land-use planning policies. To build this model, it was necessary to take into account a variety of factors, each of them available as either datasets or specialized (sub)models and the only viable solution for designing the whole model in a methodological and validated way was to progressively coupling them and evaluate, at each step, the relevance of its results.

6.1 Objective

This chapter aims at demonstrating the capability of the co-modeling approach to support modelers methodologically, in the incremental design of complex models. It is based on a concrete case study where an important number of models needed to be coupled and carefully evaluated to build a complete and realistic land-use change model. I will first describe the context of this model construction and define, one by one, the different micro-models that were designed to represent different factors known to influence land-use change decisions. I will then show how these micro-models were progressively coupled thanks to the flexibility offered by the co-modeling approach.

6.2 Modeling context

The region of the Vietnamese Mekong Delta (VMD), composed of 13 provinces including one municipality and home of approximately 18 millions of inhabitants, was by far the most productive region of Vietnam in agriculture and aquaculture in 2014. In terms of rice production, for instance, 47% of the cultivated areas in Vietnam were situated in the VMD, and they produced 54% of the total production; in terms of aquaculture, 2/3 of the Vietnamese production originated from the VMD. According to [88], these performances have fueled the boom of the Vietnamese exports of agricultural products.

This growth has logically been accompanied by a deep transformation of the agricultural land-use. However, other factors, like the sea level rise, the urbanization of the region or the progression of soil salinity ([89]), have played an important role in this transformation, and it is not trivial to sort out its different causes. In a country like Vietnam, this difficulty raises some concerns because agricultural land-use is traditionally strictly planned under the control, and following the national circulars, of the Ministry of Natural Resources and

Environment ([90], [91]). Plans are produced every ten years and readjusted at mid-term using a land-use inventory in order to rectify divergences with the reality. This actually results in two five-years long plans, detailed down to the level of provinces, that both recommend a given distribution in terms of land-use and cultivation types, but also schedule national and provincial investments (irrigation infrastructures, protection against flooding or salinity intrusion, transportation infrastructures, and so on) based on this distribution. In an ideal situation, where every province would follow the plan, there would not be any difference between the recommended distribution of land-use and its forecast. However, during the period covered by the latest plan (from 2000 to 2010), the planned – and then expected – distribution has been systematically offset, sometimes by an important margin, from the reality of land-use as measured by remote-sensing techniques. In Figure 6.1, for instance, it is possible to see that land-use has had a trend to shift from rice to shrimps. The surface dedicated to rice crops has strongly decreased, while the one dedicated to shrimp aquaculture has increased.

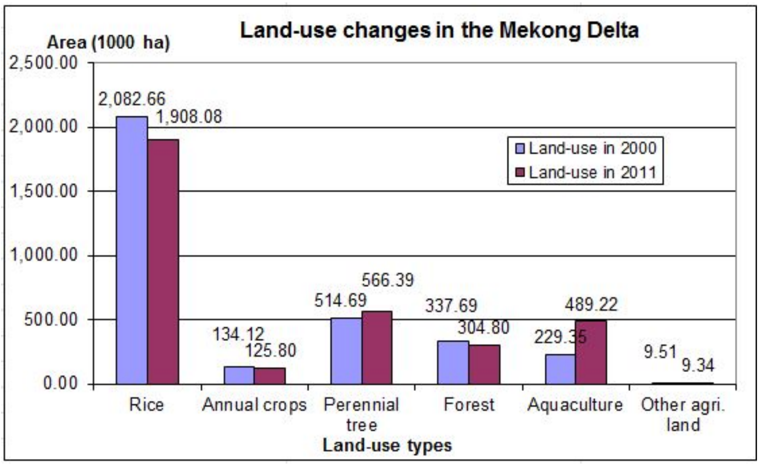

Figure 6.1: Land-use area in the Mekong Delta in 2000 and in 2011. *Source: Vietnamese General Statistics Office ([92]) and Ministry of Natural Resources and Environment ([93]) of Vietnam*

To understand these changes, we consider in this chapter a specific case study, which will be the focus of the remaining sections. This case study comprises 5 villages situated in the middle of the Thanh Phu district (Ben Tre province). They have been carefully chosen as they exhibit a variety of land cover characteristics while remaining geographically close to each other, at least close enough to reasonably allow us to consider that the farmers living in these villages share common "cultural traits" and traditions.

In Figure 6.2, we show the results of a study conducted on these 5 villages in order to assess the shift of land-use between, on one hand, the two projections for year 2010 of the plans produced in 2000 and 2005 and, on the other hand, the actual land-use map in 2010 ([94]). Changes are measured using a Fuzzy Kappa indicator ([95]), a variant of Kappa ([96]) that provides a measure close to how humans compare maps. The darkness of the areas on the two right-hand maps is proportional to the change in land-use. It is easy to see that, while the average changes for the whole province may not be spectacular, they translate into local changes that mark complete shifts from one type of production to another. With respect to this, the plan published in 2000 is completely wrong in its projections (almost all parcels have changed) and the rectified plan published in 2005, while correct for the most part, completely misses the shifts in two villages and along the canals.

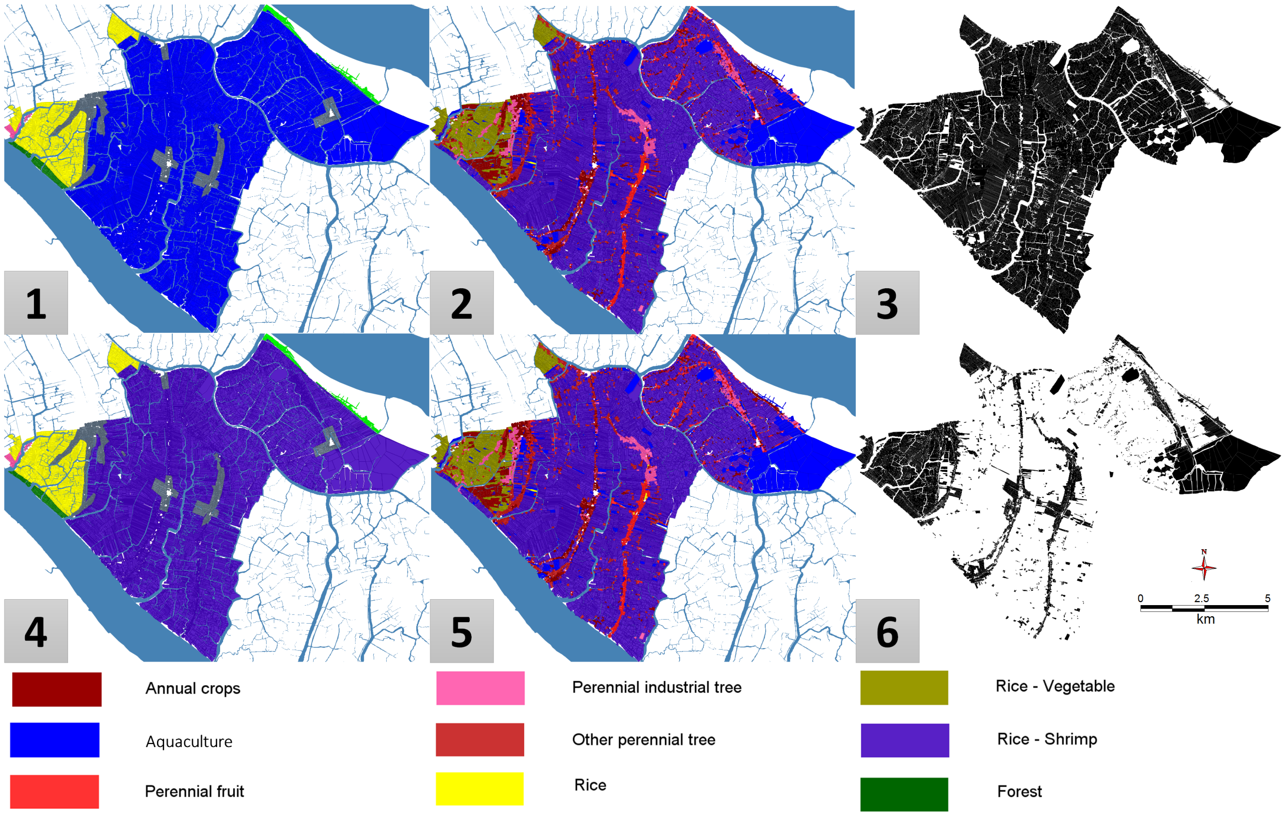

Figure 6.2: Comparison of land-use planning map and land-use map in 2010 of 5 villages of Thanh Phu district, Ben Tre province. (1) Land-use planning map 2010 (planned in 2001), (4) Land-use planning map 2010 (modified in 2005)), (2)(5) Land-use map in 2010, (3) Fuzzy Kappa map between 1 and 2, (6) Fuzzy Kappa map between 4 and 5

As pointed by [97] and [98], the dynamics of land-use change at a regional scale results from the interactions of various actors and factors at different scales, among them institutional policies, individual farming choices, land-cover and environmental changes, economic conditions, social dynamics, etc. To understand their interactions and respective influences, modelers need to represent the individual contribution of each of these factors, possibly in different focused models, and to specify how these models interact to produce the emergent phenomena observed above.

6.3 Definition of micro-models

Firstly, we represent the models that have been used in this project and that have been progressively coupled to form the complete integrated model. Several models had to be defined, including:

- M_F: a cognitive model representing the individual farmers and their decision-making processes using a BDI formalism;
- M_S: an environmental model representing the intrusion of salted water in the delta and the diffusion of salinity in the soils;
- M_P: a land suitability model based on spatial analyses and GIS and remote sensing data;
- M_E: an economic model representing the evolution of agricultural products prices at the regional scale;
- M_N: and finally a social network model representing the relationships of influence between farmers.

  These models are presented with some more details below, and the reader willing to have a complete description of each of them should consult either [99] [100]

6.3.1 Farmers behaviors model (M_F)

To represent the quite complex behavior of farmers and especially their decision-making processes about what type of land-use they should apply to their parcel, we reuse the existing farmer decision-making model of [101]. In this model, the authors have compared three formalisms (i.e., BDI, multi-criteria and probabilistic) for representing complex individual decision-making processes in agent-based models and arrived to the conclusion that BDI is the most relevant when dealing with heterogeneous factors such as the ones we consider in our integrated model.

6.3.2 Salinity intrusion model (M_S)

We designed a simple model exclusively dedicated to the reproduction of the dynamics of soil salinity from 2005 to 2010. As shown in Figure 6.3, the inputs of this model are the salinity maps of 2005 [102] and 2010, available thanks to an efficient regional network, the GIS data on dikes and dike-protected areas for the year 2010 ([94]), the GIS data on parcels and their land-use and the GIS data on rivers and canals.

This model relies on a discretization of the environment in 18400 parcels, obtained from the land-use map, where each parcel is linked with its immediate neighbors in a radius of 100m and is provided with a set of attributes, among them its salinity (classified into 4 levels (less than 0.4%; 0.4 - 0.8%; 0.8 - 1.2%; greater than 1.2%)) whether or not it is in dike-protected area, and whether or not it is bordering a river (obtained by overlapping the rivers and canals maps). Initial salinity levels in Thanh Phu district, in 2005, are computed after ([102]).

The dynamics of the model is voluntarily kept simple and deterministic: at each iteration (one year) it reevaluates, like in a cellular automaton, the level of salinity of each parcel. Parcels considered as protected by dikes do not change. Parcels bordering rivers see their salinity automatically rise up to 1.2%. And salinity is diffused in the remaining parcels using the following function:

$$salinity(x) = \frac{salinity(x) + \sum salinity(y|distance(x,y) <= 100)}{1 + |salinity(y|distance(x,y) <= 100)|} with x, y \in Parcels \tag{6.1}$$

One major limitation of this model is that we did not consider the various flooding episodes that occurred during these five years, principally to keep it as generic as possible, and also because of the lack of accurate data on these episodes. Taking them into account would probably require the use of a stochastic component, which could in any case be added later if necessary.

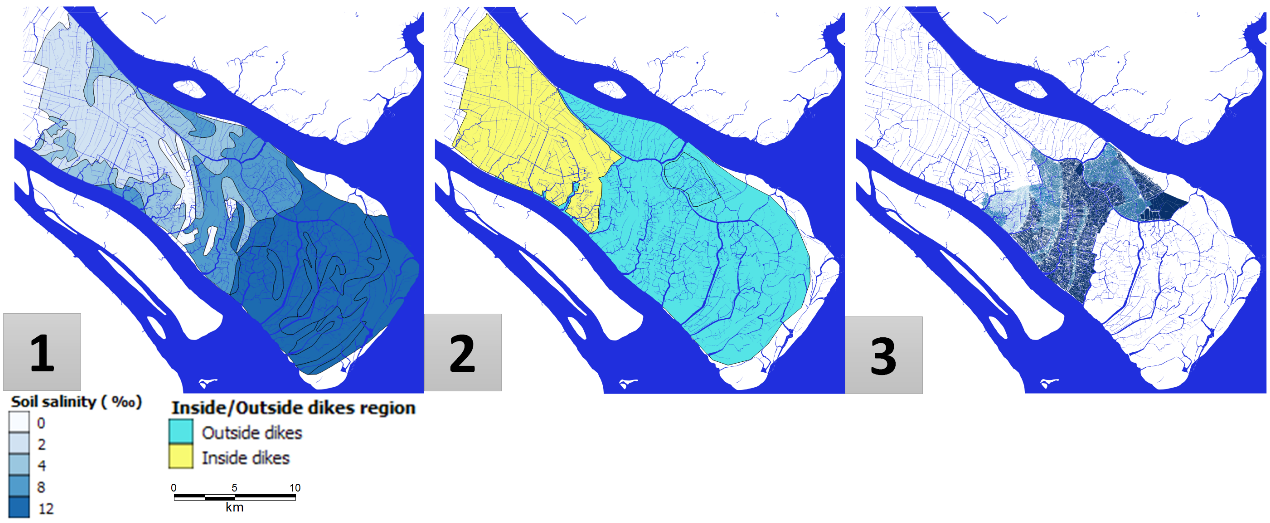

Figure 6.3: Data for the soil salinity model: (1) Soil salinity map in 2005, (2) Regions protected by dikes in 2010, (3) Soil salinity map in 2010

6.3.3 Parcels model (M_P)

The environment in which land-use changes are simulated is represented by a set of parcel agents, initialized after a land-use map at the level of villages (Figure 6.4). By combining this map with a soil map and a flooding map, each parcel agent is provided with a given land-use and other attributes such as its soil type, its level of salinity, and the extent and depth of flooding episodes on it. The two main purposes of this sub-model are, on one hand, to provide other sub-models with a unified way of accessing and modifying these attributes and, on the other hand, to compute a synthetic indicator called "land suitability". Land suitability represents the compatibility of a given parcel with the different land-use types. It can take 4 values (S1: Highly suitable; S2: Moderately suitable; S3: Marginally suitable ; N: Not suitable) ([103]). Based on the type of soil and the level of salinity, we defined (with the help of domain experts) a suitability matrix for each of the 8 land-use types considered in the model (e.g. Rice, Rice-Vegetables, Rice-Shrimp, Shrimp, Annual crops, Industrial perennial, Fruit perennial and Other perennial).

In the current instantiation of the integrated model, this sub-model is not provided with any internal dynamics. Instead, it is supposed that some attributes can be manipulated by external models (e.g., M_S for the level of salinity) and that the type of the soil remains unchanged. Each year, each of the parcels then simply computes and updates its land suitability matrix.

6.3.4 Economical model (M_E)

We then designed a simple economic model to represent the evolution of the regional market prices and costs of production. The data concerning market prices has been collected from 2005 to 2010 (averaged every year) from regional sources. However, the costs of production within the corresponding period could not be obtained so easily; we then used the costs in 2010 (evaluated in

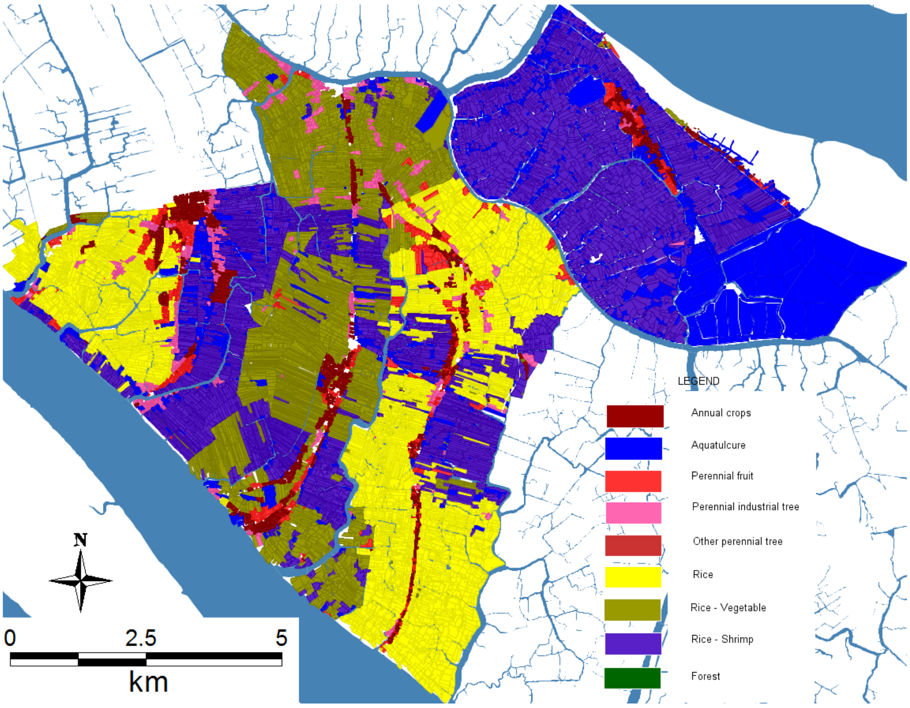

Figure 6.4: Land-use map of five villages (An Thanh, Binh Thanh, An Thuan, An Quy, An Nhon, An Dien) of Thanh Phu district in 2005

[104]) and extrapolated them from 2005 to 2010 using the regression equations ([105]) depicted in Figure 6.5:

The main components of the model are the 5 regression equations below, where 'x' represents the time in year from 1 to 5 (i.e. from 2005 to 2010) and the parameters have been computed after the values for 2010. The costs are expressed in the Vietnamese currency, Dongs, per square meter (VND/m2). Such a model allows us to easily compute the expected benefit of a given production, by subtracting its cost from its selling price, multiplied by the surface of the parcel on which it is cultivated.

$$cost\_vegetable = (1226.4x - 917.55) \tag{6.2}$$

$$cost\_coconut\_fruit = (1304.5x + 910.91) \tag{6.3}$$

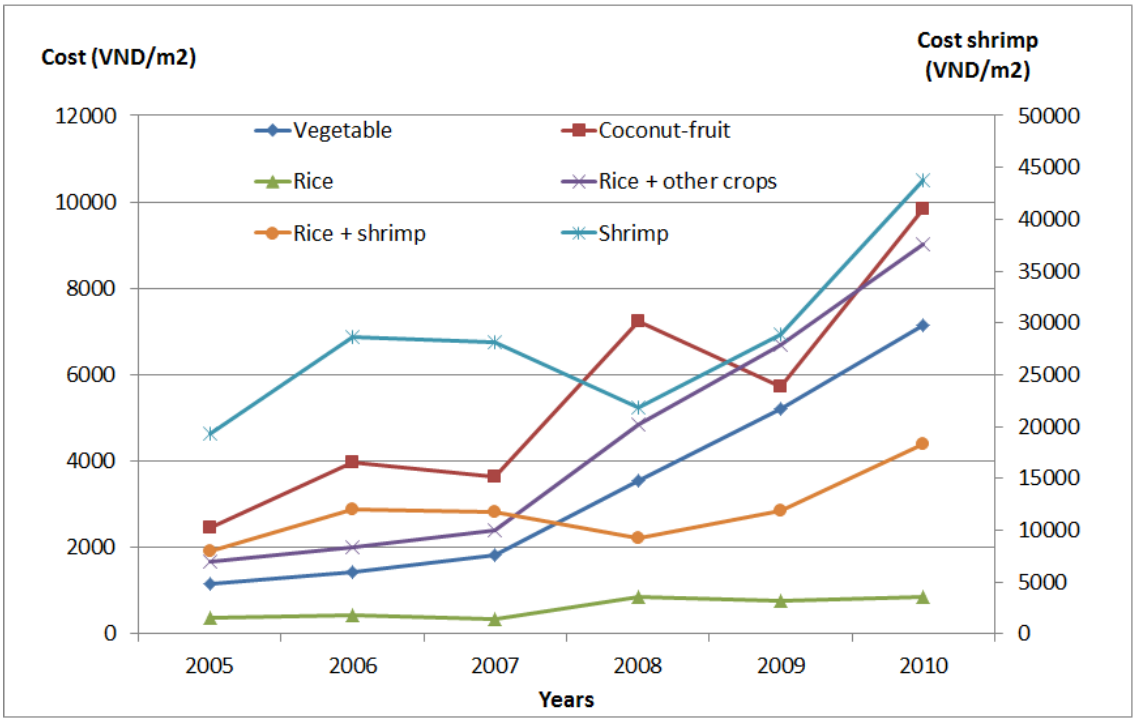

Figure 6.5: Production costs of the most popular products in the Mekong Delta from 2005 to 2010. *Source: Computed from the prices from 2005 to 2010 and the production costs in 2010 (1USD ~21,840VND)*

$$cost\_rice = (-17.71x^3 + 189.3x^2 - 471.95x + 688.9) \tag{6.4}$$

$$cost\_rice\_other = (1519.1x - 880.31) \tag{6.5}$$

$$cost\_shrimp = (1345.2x^3 - 13094x^2 + 38752x - 7459.8) \tag{6.6}$$

$$cost\_rice\_shrimp = (137.74x^3 - 1345.6x^2 + 3998.2x - 865.13) \tag{6.7}$$

6.3.5 Farmers relationships model (M_N)

In [106], the author supposes the existence of a network in which farmers can be influenced by and can influence their "neighbors". This concept of "neighborhood" can take many forms, from topological or geographical relationships, which rely on the proximity between farmers, to familial or socio-economic ones, in which, for instance, the level of income would be used as a filter. A first assumption is made here by considering that the familial network is superseded by the proximity network since in Vietnam, especially in rural areas, it is common that members of the same family live next to each other. A second assumption is that the exchanges of influence take place between farmers that belong to the same "social level" (or income group).

Statistical population data used at the provincial level ([107]) distinguishes between 3 different profiles of farmers, essentially based on their level of income: (1) P1: rich and standard farmers, (2) P2: average farmers, (3) P3: poor and nearly poor farmers. We reuse this classification and couple it with the proximity network in order to produce an "influence network" for each farmer.

This network is recomputed at every iteration of the simulation (as farmers may change their income) and its main purpose is to serve as a "social topology" for farmers, i.e. to modify the way they compute their set of neighbors. In the absence of this sub-model, the neighbors of a farmer are the farmers located in a radius of 100m around it. When this sub-model is used, the neighbors become the farmers located in the same radius and belonging to the same profile.

6.3.6 Summary on the micro-models

The five models presented above are completely heterogeneous, be it in terms of modeling formalism, data required or scientific domain. Implementing and testing them all at once in one single model would certainly result in a complex construction difficult to design, maintain, adapt and experiment. The co-modeling approach provides a solution to this problem: as models, we can test and calibrate them individually, but, as micro-models belonging to the same co-model, we can test their respective influences in various patterns of composition. And, once each of them and specific combinations of them have been calibrated and validated, they can be used to form the complete integrated model. To this aim, I will present in the three following sections the incremental path taken to build the complete model and show that it provides modelers with much more flexibility than existing approaches.

6.4 Impact of environmental factors on farmers' decisions

Environmental characteristics, such as saltwater intrusion, soil type, etc., change the suitability of parcels with respect to agricultural products and land-use types. In consequence, not all land-use types can be applied to a given parcel. This directly or indirectly influences the strategy of farmers in reality, and the existence of this influence needed to be tested in our model. This was realized by designing a first co-model that only coupled the parcels, salinity and farmers micro-models.

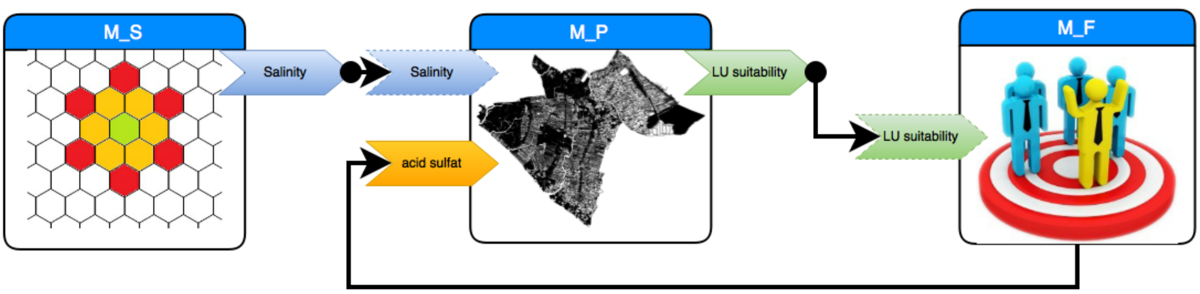

Figure 6.6: Conceptual view of the three micro-models coupled in the "Impact of environmental factors" model

6.4.1 Implementation

Step 1

Firstly, we consider the Farmer model as M_F , the Parcel model as M_P and the salt intrusion model as M_S. And we import them in a newly created co-model

```
import "Farmer.gaml" as M_F
import "ParcelDynamic.gaml" as M_P
import "EnvironmentCA.gaml" as M_S
```

Step 2

Secondly, the micro-models are instantiated using one of their experiments (each of them having only one). In addition, we also set the time scale of the simulation to be 1 month in order to align all the micro-models on the same temporal scale.

```
import "Farmer.gaml" as M_F
import "ParcelDynamic.gaml" as M_P
import "EnvironmentCA.gaml" as M_S

global{
    float step<-1 #month;
    init{
        create M_F.simpleBDI;
        create M_P.simpleParcel;
        create M_S.simpleEnv;
    }
}
```

Step 3

The coupling mechanism consists firstly (and simply), at each step of the co-model, in asking each micro-model to do a number of steps corresponding to its own temporal scale.

```
import "Farmer.gaml" as M_F
import "ParcelDynamic.gaml" as M_P
import "EnvironmentCA.gaml" as M_S

global{
```

```
    float step<-1 #month;
    init{
       create M_F.simpleBDI;
       create M_P.simpleParcel;
       create M_S.simpleEnv;
    }

    reflex doCoSimulation{
       ask M_S.simpleEnv     {loop times:6 {do _step_;}}
       ask M_P.simpleParcel     {loop times:1 {do _step_;}}
       ask M_F.simpleBDI     {loop times:2 {do _step_;}}
    }
}
```

Step 4

As described in the data flow on Figure 6.6, the model M_F receives the suitability attribute of M_P as and input while M_P take acid_sulfat of M_F and the salinity from M_S (lines 18-21). The data flow is described using the accessors defined for each micro-model (or their wrapper).

```
1  model Comodel
2  import "Farmer.gaml" as M_F
3  import "ParcelDynamic.gaml" as M_P
4  import "EnvironmentCA.gaml" as M_S
5
6  global{
7      float step<-1 #month;
8      init{
9         create M_F.simpleBDI;
10        create M_P.simpleParcel;
11        create M_S.simpleEnv;
12     }
13
14     reflex doCoSimulation{
15        ask M_S.simpleEnv   {loop times:6 {do _step_;}}
16        ask M_P.simpleParcel   {loop times:1 {do _step_;}}
17
18        M_F.simpleBDI.set_land_suitability(
19           M_P.simpleParcel.get_land_suitability(
20              M_F.simpleBDI.get_acid_sulfat(),
21              M_S.simpleEnv.get_salinity_of_parcels()));
22
23
24
25        ask M_F.simpleBDI   {loop times:2 {do _step_;}}
26     }
27 }
28 experiment simple type:gui;
```

6.5 Coupling of Farmer and Socio-Economy factors

In [108] [106], the authors point out that the socio-economical factors, such as the price of products, costs of production and benefits expected are important factors influencing the decision of farmers. Farmers usually tend to produce products that are supposed to provide them with the highest income in the future. It is supposed that the economic information are shared by everyone (in the newspapers, in the markets) but they can also be transmitted using less conventional means through social interactions, between neighbors, within families or more extended social networks, and they exert an influence on the decisions of farmers. To verify that is was the case in the model as well, we repeated the previous steps to couple, this time, the economic model (M_E), the social network model (M_N) and the farmers model (M_F) (Figure 6.7)

6.5.1 Implementation

Step 1

A new co-model is initialized and imports the three micro-models previously defined.

```
model Comodel
import "Farmer.gaml" as M_F
import "MarketPrice.gaml" as M_E
import "social_eco_simple.gaml" as M_N
```

Step 2+3+4

Next, we respectively instantiate the three micro-models, still aligning them to a time step of 1 month. The coupling mechanism consists in asking each of them

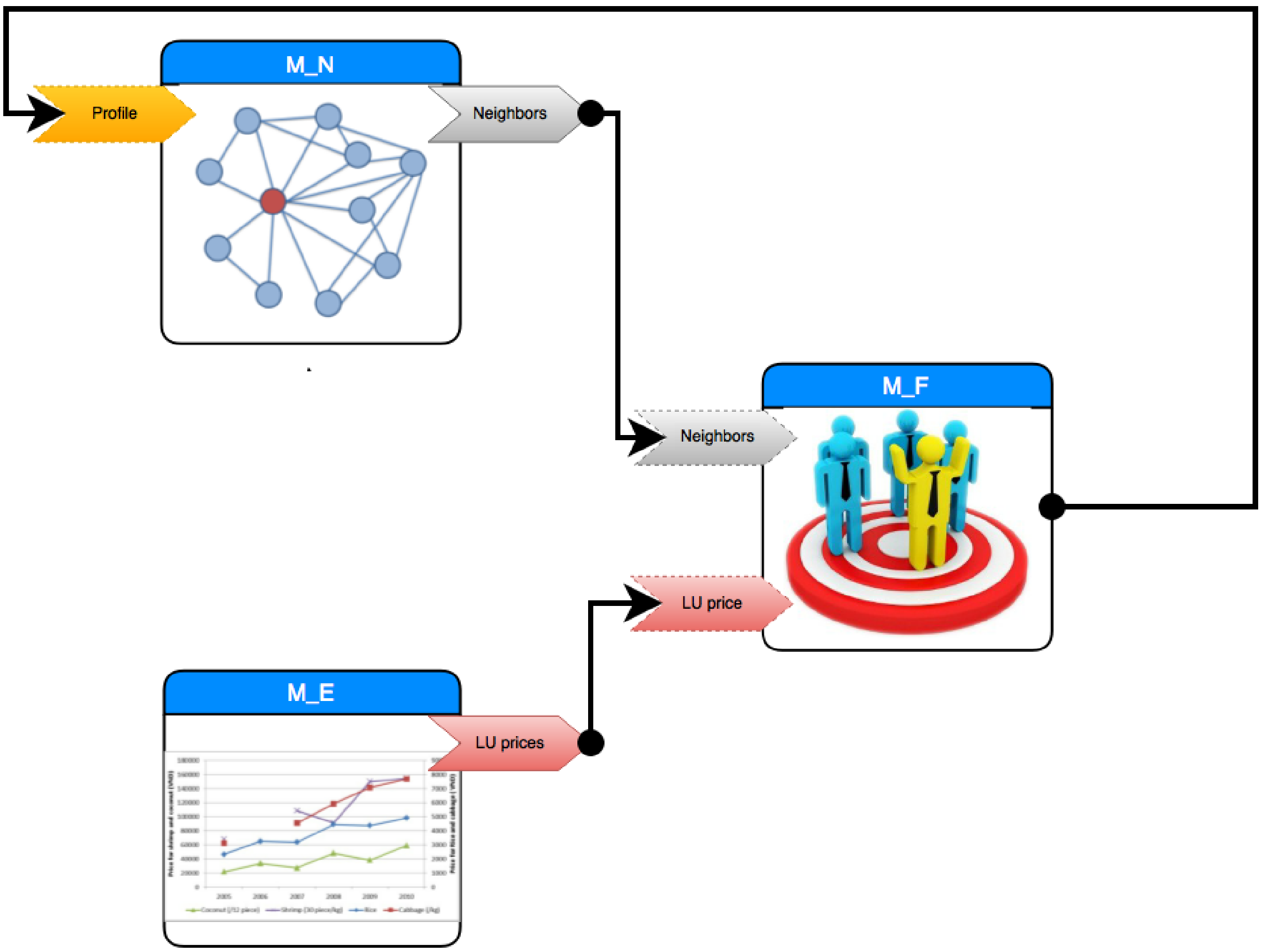

Figure 6.7: Diagram of the "Socio-Economic Factors" co-model

to step and implementing the data flow presented in 6.7, still using the accessors of the micro-models (or their wrappers).

```
model Comodel
import "Farmer.gaml" as M_F
import "MarketPrice.gaml" as M_E
import "social_eco_simple.gaml" as M_N

global{
   float step<-1 #month;
   init{
      create M_F.simpleBDI;
      create M_E.simpleMarket;
      create M_N.simpleSocial;
   }

   reflex doCoSimulation{

      if(restart_M_E) {ask M_E.simMarket {do die;} create M_E.simMarket;}
      ask M_E.simMarket    {loop times:1 {do _step_;}}
      M_F.simpleBDI.set_cost(M_E.simMarket.get_cost());

      if(restart_M_N) {ask M_N.simSocial {do die;}  create M_N.simSocial;}
      M_N.simSocial.set)profiles(M_F.simpleBDI.get_profiles());
      ask M_N.simSocial     {loop times:1 {do _step_;}}
      M_F.simpleBDI.set_neighbors(M_N.simSocial.get_neighbors());

      ask M_F.simpleBDI     {loop times:2 {do _step_;}}
   }
}
experiment simple type:gui;
```

6.6 Coupling environmental, social and economic models

The results corresponding to the first 2 co-models can be consulted in [99]. Once these two models had been calibrated and validated with respect to the data available, we could move on to the complete integration of all the micro-models, in order to represent the (realistic) situation where farmers agents are influenced by both environmental factors and socio-economic ones. The conceptual diagram (figure 6.8) that describes the wrapping and the coupling of the models listed above is, not surprisingly, a simple merge of the two previous diagrams depicted in figure 6.6 and 6.7 with the same data flows between micro-models.

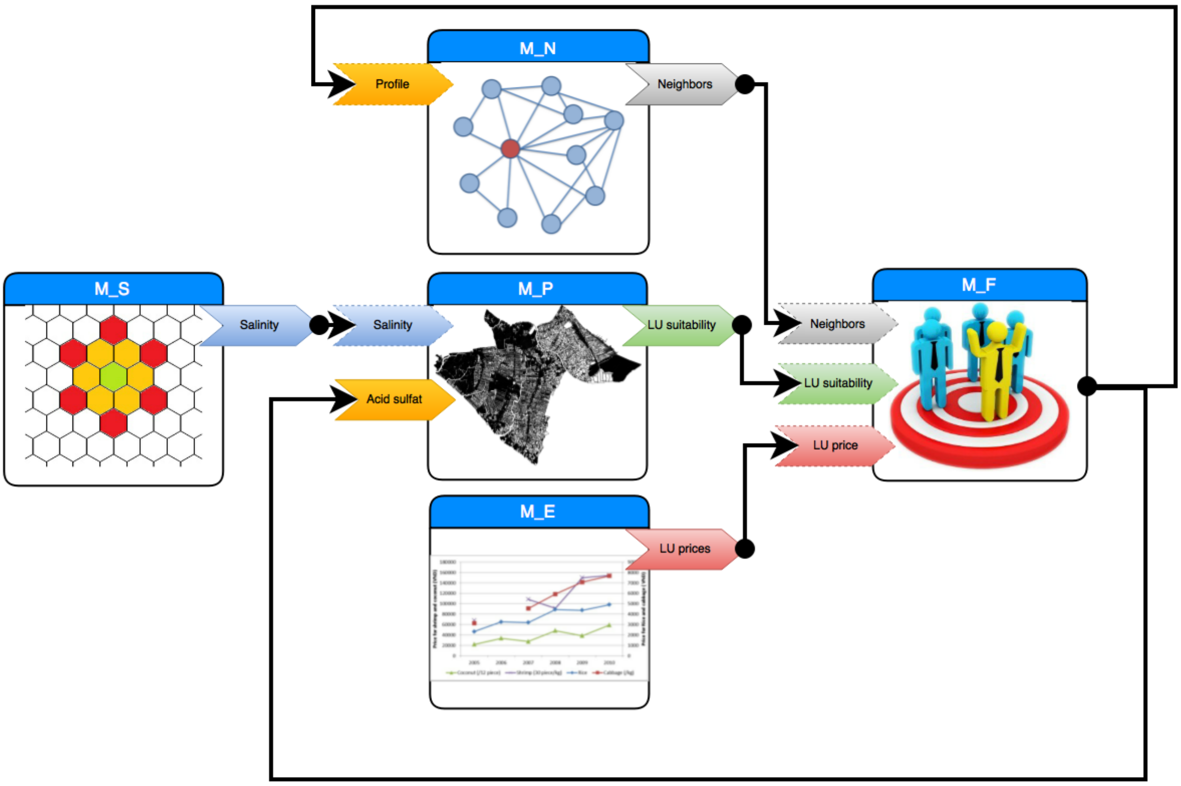

Figure 6.8: Conceptual view of the five micro-models that constitute the complete integrated model

6.6.1 Implementation

Step 1+2+3+4

Quite intuitively, the merging of the two diagrams produces a co-model that can be implemented by a straightforward combination of code of the 2 Previous co-models described above. Nothing need to be changed (except that the farmers model is of course instantiated only once in this new co-model).

```

model Comodel
import "Farmer.gaml" as M_F
import "MarketPrice.gaml" as M_E
import "ParcelDynamic.gaml" as M_P
import "EnvironmentCA.gaml" as M_S
import "social_eco_simple.gaml" as M_N

```

```
global{
   float step<-1 #month;
   init{
      create M_F.simpleBDI;
      create M_E.simpleMarket;
      create M_P.simpleParcel;
      create M_S.simpleEnv;
      create M_N.simpleSocial;
   }

   reflex doCoSimulation{


      if(restart_M_E) {ask M_E.simMarket {do die;} create M_E.simMarket;}
      ask M_E.simMarket   {loop times:1 {do _step_;}}
      M_F.simpleBDI.set_cost(M_E.simMarket.get_cost());



      if(restart_M_N) {ask M_N.simSocial {do die;}  create M_N.simSocial;}
      M_N.simSocial.set)profiles(M_F.simpleBDI.get_profiles());
      ask M_N.social     {loop times:1 {do _step_;}}
      M_F.simpleBDI.set_neighbors(M_N.simSocial.get_neighbors());


      ask M_S.simEnv     {loop times:6 {do _step_;}}
      ask M_P.simParcel   {loop times:1 {do _step_;}}
      M_F.simpleBDI.set_land_suitability(
         M_P.simParcel.get_land_suitability(
            M_F.simpleBDI.get_acid_sulfat(),
            M_S.simEnv.get_salinity_of_parcels));



      ask M_F.simpleBDI     {loop times:2 {do _step_;}}
   }
}
experiment simple type:gui;
```

6.6.2 Experimentation

Although my main interest in this project was methodological, it is interesting to see that each of the co-models described in the previous sections has been experimented and that their outputs could be compared, allowing to test the influence or importance of each set of factors on the outcomes of their simulations.

The outcome of a simulation in that case is a map of land-use in 2010. To assess its validity, two indicators are used for comparing it to the land-use map observed in 2010: Absolute Deviation Percentages (ADP), which measures the global absolute difference between the maps, and Fuzzy Kappa (FKappa [95]), used to measure their similarity based on local correlations. These indicators are already implemented in the GAMA platform.

$$ADP(\%) = 100\frac{\sum_{i=1}^{n}|\widehat{X}_i - X_i|}{\sum_{i=1}^{n}\widehat{X}_i} \tag{6.8}$$

with: $\widehat{X}_i$ the observed quantity of parcels with land-use i and X_i the simulated quantity of parcels with land-use i.

Because of the possible stochasticity of some of these combinations, 100 simulations are launched for every experiment. Their ADP and Fuzzy Kappa are computed, and we use a one-way ANOVA with the assumption of equal variances and a 95% confidence interval ([109])) to produce their average values.

The experiments conducted on the first co-model produced a FKappa at 39.4% and an ADP at 43.22% , which increased to almost 43.00%, while the ADP sharply decreases to 31.47% for the second co-model. Finally, the experiments on the complete co-model revealed an ADP of 22% (meaning a global accuracy of 78% in terms of surfaces devoted to each land-use) and a FKappa of 47.92%. The experimental results are shown on figure 6.9 (where a highest FKappa and a lowest ADP, which indicate the most realistic outcomes, are obtained for the complete co-model).

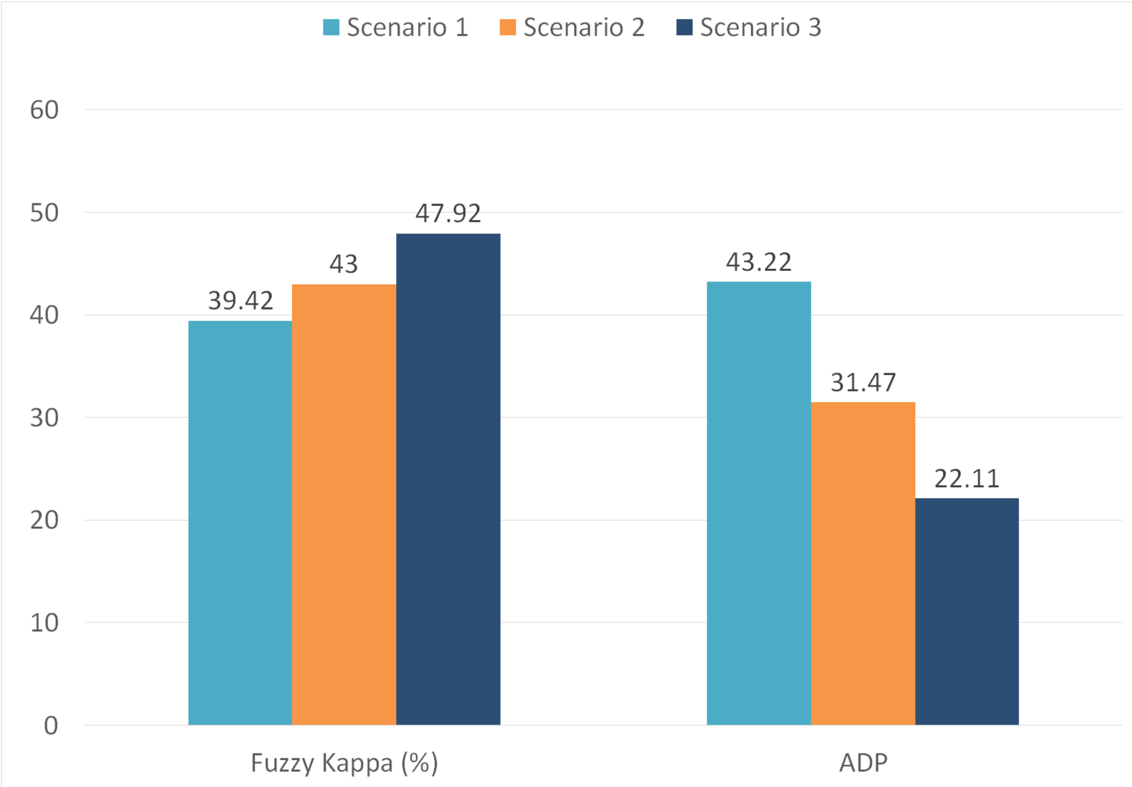

Figure 6.9: The average result of 100 simulation using the Fuzzy Kappa measurement.

6.7 Conclusion

By considering each micro-model as an agent, co-modeling definitely eased the experimental process depicted above, allowing us to add or remove micro-models with a minimum of efforts (a few code changes, and only in the co-model, not in any of the micro-models). Moreover, we have been able to manipulate, without any particular problem, models defined in different formalisms and make them communicate and exchange data and control in a completely transparent way. This experimental flexibility has proved precious, as well, to progressively refine and test either the integrated model or the existing micro-models against real data. This clearly demonstrates the usability of the co-modeling framework for progressively optimizing the representation provided by an integrated model, adding, removing or adapting, according to the simulation outcomes, the factors that have been envisioned at the beginning.

Between the three co-models, the only changes required have concerned the "glue" between micro-models (i.e. the coupling mechanism implemented inside the co-model itself), and we have seen as well that is has remained very simple to read and very modular: merging two co-models simply resulted in the combination of their coupling mechanisms.

Chapter 7

Conclusion

This manuscript has presented an agent-based approach for supporting modelers in designing, building and experimenting complex integrated models, composed of several coupled micro-models. This proposal, named “co-modeling”, distinguishes itself from similar proposals in the literature by the simplicity of its concepts (immediately understandable by anyone working on agent-based modeling) and the fact that is has been fully implemented and tested in a popular modeling and simulation platform. It also fulfills four requirements that existing proposals have difficulties to address: **reusability**, in that it allows to easily reuse legacy models in different co-models but also to reuse coupling patterns defined in co-models in other modeling contexts; **expressivity**, in that the coupling mechanisms are both entirely transparent and writable in an agent-based modeling language, requiring no further efforts from modelers; **scalability**, in that, although it does not provide any magic solution to the various problems of scale transfer, it allows to take into account and adapt the spatial and temporal boundaries of micro-models when they are coupled; **flexibility**, finally, in that micro-models can be assembled and disassembled not only at design-time, but also during simulations, allowing to test their influence on the integrated model, to switch between them when it is necessary, and to easily integrate data sources along with models. It may be the case that what I presented in this document, especially the experimental validation on selected case studies, could be done using other existing technologies, especially the ones emanating from software engineering and object-oriented (or component-oriented) approaches. I do not deny it, and neither do I deny the strong influence of some existing agent-based architectures (especially the one proposed in Madkit) on my proposal. It is, however, the first time, to my knowledge, that the possibility of coupling models in such an easy way is offered to the "average modeler" in a ready to use package, with no other requirement than being able to write an agent-based model (a skill which is more and more common in different scientific domains). It may also be the case that modelers find this proposal too "weak" in that it does not provide solutions to the problems of scale transfer, aggregation/disaggregation, data exchanges, or control flows between models. Instead, I advocate the fact, in this thesis, that there doesn't exist any "general" solution to these problems and that the only relevant contribution to solve them is not in providing ready-made recipes that will probably fail for a particular integration of models but to provide modelers with tools (concepts, languages, operational architectures) that allow them to express these problems in the most transparent and explicit way. The contribution of my research work is then principally this operational architecture and its accompanying supports (concepts, language), all of them being available and immediately usable (and already used in multiple projects) for coupling heterogeneous models. It is easy to see, however, that such a contribution opens a lot more perspectives than the ones already presented in this document and requires, to be fully adopted, a number of developments that I was not able to handle during these 4 years. Among the developments that appear to me as the most urgent, I can list:

- The development of the co-modeling architecture on other simulation platforms and modeling languages, as long as they support agent-based multi-level modeling abstractions (which is not the case in all).
- The development of abstract "coupling patterns", reusable and instantiable in different modeling contexts, which could serve as the basis of a more general methodology for coupling models depending on the goal(s) of the modeler (for instance, patterns for switching between models, for building incremental models, etc.) - The formalization of the "wrapper models" into something that can be described – and tested at simulation-time – in a co-model (i.e. like classes can specify the interface of objects they can manipulate, for instance, in Java)
- The development of a more intuitive tool, maybe based on a graphical interface, to define the coupling of models and their interactions. The work done in VLE regarding DEVS models would be an excellent starting point for this.
- The translation to and from the specifications produced by HLA and FMI. This point is particularly important as it could represent an interesting bridge between my proposal and more mature ones, which nevertheless fail to be adopted by the modeling community. The concepts present in

HLA can be translated without too much difficulties to the definition of a co-model (where Federations, Assemblies would be micro-models and the RTI the coupling mechanism), while the specifications of FMI could be potentially a good way to define wrapper models.

In any case, this PhD thesis has been an incredible journey, during 4 years, in the multidisciplinary world of modeling and simulation and I hope that my work will, even at its modest scale, contribute to this domain and help modelers in building better models.

Bibliography

[1] K. Smith, H. Brighton, and S. Kirby, "Complex Systems in Language Evolution: The Cultural Emergence of Compositional Structure," *Advances in Complex Systems*, vol. 06, pp. 537–558, Dec. 2003.

[2] H. H. Pattee, *Hierarchy Theory; The Challenge of Complex Systems.* New York: George Braziller, June 1973.

[3] L. M. Rocha, "Complex systems modeling: Using metaphors from nature in simulation and scientific models," *BITS: Computer and Communications News*, 1999.

[4] K. Kiyoto, "A Comparative Study for Risk Management Industrialization between Thailand and Vietnam-Risk perceptions to the recent floods-," *Japan Regional Annual Meeting Scientific Papers Collection (CD-ROM)*, vol. 49th, pp. 5–3, 2012.

[5] N. T. N. Anh, Z. J. Daniel, N. H. Du, A. Drogoul, and V. D. An, "A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks," in *Advanced Agent Technology* (F. Dechesne, H. Hattori, A. t. Mors, J. M. Such, D. Weyns, and F. Dignum, eds.), no. 7068 in Lecture Notes in Computer Science, pp. 371–383, Springer Berlin Heidelberg, May 2011. DOI: 10.1007/978-3-642-27216-5_28.

[6] T. N. A. Nguyen, Y. Chevaleyre, and J. D. Zucker, "Optimizing the Placement of Evacuation Signs on Road Network with Time and Casualties in Case of a Tsunami," in *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 394–396, June 2012.

[7] T. N. A. Nguyen, J. D. Zucker, M. H. Nguyen, A. Drogoul, and H. P. Nguyen, "Simulation of Emergency Evacuation of Pedestrians along the Road Networks in Nhatrang City," in *2012 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pp. 1–6, Feb. 2012.

[8] V.-M. Le, Y. Chevaleyre, J.-D. Zucker, and Ho Tuong Vinh, "Speeding up the evaluation of casualties in multi-agent simulations with Linear Programming application to optimization of sign placement for tsunami evacuation," pp. 215–220, IEEE, Nov. 2013.

[9] V.-M. Le, Y. Chevaleyre, J.-D. Zucker, and H. Tuong Vinh, "Approaches to Optimize Local Evacuation Maps for Helping Evacuation in Case of Tsunami," in *Information Systems for Crisis Response and Management in Mediterranean Countries* (C. Hanachi, F. Bénaben, and F. Charoy, eds.), vol. 196, pp. 21–31, Cham: Springer International Publishing, 2014.

[10] V.-M. Le, Y. Chevaleyre, Ho Tuong Vinh, and J.-D. Zucker, "Hybrid of linear programming and genetic algorithm for optimizing agent-based simulation. Application to optimization of sign placement for tsunami evacuation," pp. 138–143, IEEE, Jan. 2015.

[11] V. Pham, "Things to know about Brown Plant Hopper and preventive measures," *Labour Publishing House, Viet nam (In Vietnamese)*, 2006.

[12] N. Ngo, "Principles and Applications in Mathematical model for Biological Studies, Agriculture and Environment," *Agriculture Publishing House (2008) (in Vietnamese)*, pp. 146–161, 2008.

[13] F. Horgan, "Mechanisms of resistance: a major gap in understanding planthopper-rice interactions," *Planthoppers: new threats to the sustainability of intensive rice production systems in Asia*, pp. 281–302, 2009.

[14] A. Otuka, "Migration of rice planthoppers and simulation techniques," *Planthoppers: new threats to the sustainability of intensive rice production systems in Asia*, pp. 343–356, 2009.

[15] J. L. A. Catindig, G. S. Arida, S. E. Baehaki, J. S. Bentur, L. Q. Cuong, M. Norowi, W. Rattanakarn, W. Sriratanasak, J. Xia, and Z. Lu, "Situation of planthoppers in Asia," *Planthoppers: New threats to the sustainability of intensive rice production systems in Asia*, pp. 191–220, 2009.

[16] A. Otuka, T. Watanabe, Y. Suzuki, M. Matsumura, A. Furuno, and M. Chino, "A migration analysis of the rice planthopper Nilaparvata lugens from the Philippines to East Asia with three-dimensional computer simulations," *Population Ecology*, vol. 47, pp. 143–150, Apr. 2005.

[17] N. Huan, L. Thiet, H. Chien, and K. Heong, "Farmers' participatory evaluation of reducing pesticides, fertilizers and seed rates in rice farming in

the Mekong Delta, Vietnam," *Crop Protection*, vol. 24, pp. 457–464, May 2005.

[18] K. Heong, M. Escalada, N. Huan, H. Chien, and P. Quynh, "Scaling out communication to rural farmers: lessons from the "Three Reductions, Three Gains" campaign in Vietnam," *Research to impact: case studies for natural resource management for irrigated rice in Asia*, pp. 207–220, 2010.

[19] V. Truong Xuan, *Optimization by Simulation of an Environmental Surveillance Network : application to the Fight against Rice Pests in the Mekong Delta (Vietnam)*. Paris 6, June 2014.

[20] J. S. Dahmann and K. L. Morse, "High level architecture for simulation: An update," in *Proceedings of the Second International Workshop on Distributed Interactive Simulation and Real-Time Applications*, DIS-RT '98, (Washington, DC, USA), pp. 32–, IEEE Computer Society, 1998.

[21] A. Grignard, P. Taillandier, B. Gaudou, D. A. Vo, N. Q. Huynh, and A. Drogoul, "GAMA 1.6: Advancing the art of complex Agent-Based Modeling and Simulation," in *PRIMA 2013: Principles and Practice of Multi-Agent Systems* (G. Boella, E. Elkind, B. T. R. Savarimuthu, F. Dignum, and M. K. Purvis, eds.), no. 8291 in Lecture Notes in Computer Science, pp. 117–131, Springer Berlin Heidelberg, Jan. 2013.

[22] G. N. Gilbert, *Agent-based models*. No. no. 07-153 in Quantitative applications in the social sciences, Los Angeles: Sage Publications, 2008.

[23] T. Kühne, "What is a Model?," *Dagstuhl Seminar Proceedings*, 2005.

[24] V. Grimm and S. F. Railsback, *Individual-based modeling and ecology*. 2005.

[25] Y. E. Fianyo, *Couplage de modèles à l'aide d'agents: le système OSIRIS*. PhD thesis, ANRT, Grenoble, 2001.

[26] C. H. Phan, H. X. Huynh, and A. Drogoul, "An Agent-Based Approach to the Simulation of Brown Plant Hopper (BPH) Invasions in the Mekong Delta," in *2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pp. 1–6, Nov. 2010.

[27] V. G. N. Nguyen, H. X. Huynh, T. T. Vo, and A. Drogoul, "On Weather Affecting to Brown Plant Hopper Invasion Using an Agent-based Model," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '11, (New York, NY, USA), pp. 150–157, ACM, 2011.

[28] V. X. Truong, A. Drogoul, H. X. Huynh, and M. N. Le, "Modeling the Brown Plant Hoppers Surveillance Network Using Agent-based Model: Application for the Mekong Delta Region," in *Proceedings of the Second Symposium on Information and Communication Technology*, SoICT '11, (New York, NY, USA), pp. 127–136, ACM, 2011.

[29] C. Swinerd and K. R. McNaught, "Design classes for hybrid simulations involving agent-based and system dynamics models," *Simulation Modelling Practice and Theory*, vol. 25, pp. 118–133, June 2012.

[30] C. E. Vincenot, S. Mazzoleni, and L. Parrott, "Editorial: Hybrid Solutions for the Modeling of Complex Environmental Systems," *Frontiers in Environmental Science*, vol. 4, Aug. 2016.

[31] I. Hassoumi, *Approche multi-agent de couplage de modèles pour la modélisation des systèmes complexes spatiaux: application à l'aménagement de l'espace urbain (ville de Métouia)*. PhD thesis, Paris 6, Paris 6, 2013.

[32] B. Zeigler, Y. Moon, D. Kim, and G. Ball, "The DEVS environment for high-performance modeling and simulation," *IEEE Computational Science Engineering*, vol. 4, pp. 61–71, July 1997.

[33] D. David, D. Payet, A. Botta, G. Lajoie, S. Manglou, and R. Courdier, "Un couplage de dynamiques comportementales : le modèle ds pour l'aménagement du territoire," in *JFSMA'07*, pp. 129–138, 2007.

[34] Pouliot, Jacynthe, "Définition d'un cadre géosémantique pour le couplage des modèles prévisionnels de comportement et des SIG: application pour les écosystèmes forestiers," 1999.

[35] J. C. F. Junior, *Ingénierie de systèmes d'information : une approche de multi-modélisation et de méta-modélisation*. phdthesis, Université Joseph-Fourier - Grenoble I, July 1997.

[36] M. Picavet, *La complexité dans la modélisation du système d'information de l'entreprise: propositions de solutions*. 1997.

[37] T. W. Nicolai, L. Wang, K. Nagel, and P. Waddell, "Coupling an urban simulation model with a travel model–a first sensitivity test," *Computers in Urban Planning and Urban Management (CUPUM), Lake Louise, Canada. Also VSP WP*, pp. 11–07, 2011.

[38] E. Yáñez, S. Hormazábal, C. Silva, A. Montecinos, M. A. Barbieri, A. Valdenegro, A. Órdenes, and F. Gómez, "Coupling between the environment and the pelagic resources exploited off northern Chile: ecosystem indicators and a conceptual model," 2008.

[39] A. L. Steiner, J. S. Pal, S. A. Rauscher, J. L. Bell, N. S. Diffenbaugh, A. Boone, L. C. Sloan, and F. Giorgi, "Land surface coupling in regional climate simulations of the west african monsoon," *Climate Dynamics*, vol. 33, pp. 869–892, Nov. 2009.

[40] M. Rajeevan and R. Nanjudiah, "Coupled model simulations of twentieth century climate of the indian summer monsoon," *Current Trends in Science*, pp. 537–567, 2009.

[41] E. Moreira, S. Costa, A. P. Aguiar, G. Câmara, and T. Carneiro, "Dynamical coupling of multiscale land change models," *Landscape Ecology*, vol. 24, pp. 1183–1194, Nov. 2009.

[42] S. Rochette, M. Huret, E. Rivot, and O. Le Pape, "Coupling hydrodynamic and individual-based models to simulate long-term larval supply to coastal nursery areas," *Fisheries Oceanography*, vol. 21, pp. 229–242, July 2012.

[43] V. D. Quesnel, Gauthier, "Coupling of physical models and social models: multi-modeling and simulation with VLE," *Joint Conference on Multi-Agent Modelling for Environmental Management (CABM-HEMA-SMAGET'05), Bourg Saint Maurice, France*, pp. 21–25, 2005.

[44] F. Rousseaux, E. Bocher, A. Gourlay, and G. Petit, "Toward a coupling between GIS and agent simulation: USM, an OrbisGIS extension to model urban evolution at a large scale," in *OGRS 2012 Proceedings*, pp. 206–214, Oct. 2012.

[45] N. D. Nguyen, A. Drogoul, and P. Auger, "Methodological steps and issues when deriving individual based-models from equation-based models: A case study in population dynamics," in *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents, PRIMA 2008, Hanoi, Vietnam, December 15-16, 2008. Proceedings*, pp. 295–306, 2008.

[46] H. Huang, L. Wang, X. Zhang, Y. Luo, and L. Zhao, "Coupling multi-agent model and GIS to simulate pine wood nematode disease spread in ZheJiang province, china," pp. 71430X–71430X–8, Oct. 2008.

[47] W. W. Xiaobo Li, "An Agent-Centered Multi-Formalism Modeling Framework for SoS Behavior," 2013.

[48] J. Dahmann, R. Fujimoto, and R. Weatherly, "The DoD high level architecture: an update," in *Simulation Conference Proceedings, 1998. Winter*, vol. 1, pp. 797–804 vol.1, Dec. 1998.

[49] IEEE, "IEEE standard for modeling and simulation (M&S) high level architecture (HLA) - framework and rules," *IEEE Std. 1516-2000*, pp. i –22, 2000.

[50] K. T. G. Kim, Jae-Hyun, "Proposal of High Level Architecture Extension," in *Artificial Intelligence and Simulation* (T. G. Kim, ed.), no. 3397 in Lecture Notes in Computer Science, pp. 128–137, Springer Berlin Heidelberg, Jan. 2005.

[51] D. R. C. Hill, "Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems: Second Edition by B. P. Zeigler, H. Praehofer, T. G. Kim, Academic Press, San Diego, CA, 2000.," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 1, pp. 91–92, 2002.

[52] H. Vangheluwe, "DEVS as a common denominator for multi-formalism hybrid systems modelling.," *IEEE International Symposium on Computer-Aided Control System Design (ed. Andras Varga), IEEE Computer Society Press, Anchorage, Alaska, sept. 2000*, pp. 129–134, 2000.

[53] D. R. Hild, *Discrete Event System Specification (Devs) Distributed Object Computing (Doc) Modeling and Simulation*. PhD thesis, The University of Arizona, 2000. AAI9965901.

[54] T. Blochwitz *et al.*, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models," 2011.

[55] C. Bertsch, E. Ahle, and U. Schulmeister, "The Functional Mockup Interface - seen from an industrial perspective," in *Proceedings of the 10th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, pp. 27–33, Linköping University Electronic Press; Linköpings universitet, 2014.

[56] R. S. Pressman, *Software engineering: a practitioner's approach*. New York: McGraw-Hill Higher Education, 7th ed ed., 2010. OCLC: ocn271105592.

[57] Q. C. Truong, P. Taillandier, B. Gaudou, M. Q. Vo, T. H. Nguyen, and A. Drogoul, "Exploring Agent Architectures for Farmer Behavior in Land-Use Change. A Case Study in Coastal Area of the Vietnamese Mekong Delta," in *Multi-Agent Based Simulation XVI: International Workshop, MABS 2015, Istanbul, Turkey, May 5, 2015, Revised Selected Papers* (B. Gaudou and J. S. Sichman, eds.), pp. 146–158, Cham: Springer International Publishing, 2016.

[58] "MARD - Meteorological Data by Month, Years, and Stations, Ministry of Agriculture and Rural Development."

[59] M Cristiá, D A Hollmann, and C S Frydman, "A multi-target compiler for DEVS," 2016.

[60] L. Parrott, "Hybrid modelling of complex ecological systems for decision support: Recent successes and future perspectives," *Ecological Informatics*, vol. 6, pp. 44–49, Jan. 2011.

[61] C. Bertsch, E. Ahle, and U. Schulmeister, "The functional mockup interface - seen from an industrial perspective," in *Proceedings of the 10th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, no. 96, pp. 27–33, Linköping University Electronic Press; Linköpings universitet, 2014.

[62] O. Gutknecht and J. Ferber, "The madkit agent platform architecture," in *Revised Papers from the International Workshop on Infrastructure for Multi-Agent Systems: Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, (London, UK, UK), pp. 48–55, Springer-Verlag, 2001.

[63] J. Larson, R. Jacob, and E. Ong, "The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models," *International Journal of High Performance Computing Applications*, vol. 19, pp. 277–292, Aug. 2005.

[64] R. Jacob, J. Larson, and E. Ong, "M × N Communication and Parallel Interpolation in Community Climate System Model Version 3 Using the Model Coupling Toolkit," *International Journal of High Performance Computing Applications*, vol. 19, pp. 293–307, Aug. 2005.

[65] G. Quesnel, R. Duboz, r. Ramat, and M. K. Traoré, "VLE: A Multimodeling and Simulation Environment," in *Proceedings of the 2007 Summer Computer Simulation Conference*, SCSC '07, (San Diego, CA, USA), pp. 367–374, Society for Computer Simulation International, 2007.

[66] U. Wilensky, "Netlogo," http://ccl.northwestern.edu/netlogo/, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.

[67] J. d. Lara and H. Vangheluwe, "AToM3: A Tool for Multi-formalism and Meta-modelling," in *Fundamental Approaches to Software Engineering* (R.-D. Kutsche and H. Weber, eds.), no. 2306 in Lecture Notes in Computer Science, pp. 174–188, Springer Berlin Heidelberg, Jan. 2002.

[68] T. Wittig, N. R. Jennings, and E. H. Mamdani, "ARCHON: framework for intelligent cooperation," *Intelligent Systems Engineering*, vol. 3, no. 3, pp. 168–179, 1994.

[69] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE.* Hoboken, NJ: John Wiley & Sons, Ltd, 2007.

[70] L. A. Bollinger, I. Nikolić, C. B. Davis, and G. P. Dijkema, "Multimodel Ecologies: Cultivating Model Ecosystems in Industrial Ecology," *Journal of Industrial Ecology*, vol. 19, pp. 252–263, Apr. 2015.

[71] W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman, "Normative conflict resolution in multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, pp. 124–152, Nov. 2008.

[72] J. A. Sillince, "Multi-agent conflict resolution: a computational framework for an intelligent argumentation program," *Knowledge-Based Systems*, vol. 7, pp. 75–90, June 1994.

[73] C.-Y. Lo, "Multi-Agent Conflict Coordination Using Game Bargain," *Information Technology Journal*, vol. 7, pp. 234–244, Feb. 2008.

[74] Z. Tao and Z. Zhenji, "Coordination Mechanism and Negotiation Model to Resolve Conflict of Communication Virtual Service in Information Network Corporation," *International Journal of Grid and Distributed Computing*, vol. 7, pp. 167–176, June 2014.

[75] K. Kravari and N. Bassiliades, "A Survey of Agent Platforms," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 1, 2015.

[76] D.-A. Vo, A. Drogoul, and J.-D. Zucker, “An Operational Meta-Model for Handling Multiple Scales in Agent-Based Simulations,” pp. 1–6, IEEE, Feb. 2012.

[77] D.-A. Vo, A. Drogoul, and J.-D. Zucker, “Multi-level agent-based modeling: a generic approach and an implementation.,” in *KES-AMSTA* (D. Barbucha, M. T. Le, R. J. Howlett, and L. C. Jain, eds.), vol. 252 of *Frontiers in Artificial Intelligence and Applications*, pp. 91–101, IOS Press, 2013.

[78] L. Apedaille, H. Freedman, S. Schilizzi, and M. Solomonovich, “Equilibria and dynamics in an economic predator-prey model of agriculture,” *Mathematical and Computer Modelling*, vol. 19, pp. 1–15, June 1994.

[79] M. T. Johnson and A. A. Agrawal, “The ecological play of predator–prey dynamics in an evolutionary theatre,” *Trends in Ecology & Evolution*, vol. 18, pp. 549–551, Nov. 2003.

[80] J. Durand and R. Durand, “Fitting and testing a ’predator-prey’ model,” *The Journal of Mathematical Sociology*, vol. 17, pp. 51–62, May 1992.

[81] W. O. Kermack and A. G. McKendrick, “A Contribution to the Mathematical Theory of Epidemics,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 115, pp. 700–721, Aug. 1927.

[82] N. Bacaër, “McKendrick and Kermack on epidemic modelling (1926–1927),” in *A Short History of Mathematical Population Dynamics*, pp. 89–96, London: Springer London, 2011.

[83] C. Lett, P. Verley, C. Mullon, C. Parada, T. Brochier, P. Penven, and B. Blanke, “A Lagrangian tool for modelling ichthyoplankton dynamics,” *Environmental Modelling & Software*, vol. 23, pp. 1210–1214, Sept. 2008.

[84] N. T. N. Anh, Z. J. Daniel, N. H. Du, A. Drogoul, and V. D. An, “A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks,” in *Advanced Agent Technology* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, F. Dechesne, H. Hattori, A. ter Mors, J. M. Such, D. Weyns, and F. Dignum, eds.), vol. 7068, pp. 371–383, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[85] T. A. N. Nguyen, J. Zucker, N. H. Du, A. Drogoul, and D. Vo, "A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks," in *Advanced Agent Technology - AAMAS 2011 Workshops, AMPLE, AOSE, ARMS, DOCM3AS, ITMAS, Taipei, Taiwan, May 2-6, 2011. Revised Selected Papers*, pp. 371–383, 2011.

[86] N. D. Nguyen, A. Drogoul, and P. Auger, "Methodological Steps and Issues When Deriving Individual Based-Models from Equation-Based Models: A Case Study in Population Dynamics," in *Intelligent Agents and Multi-Agent Systems* (T. D. Bui, T. V. Ho, and Q. T. Ha, eds.), vol. 5357, pp. 295–306, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[87] R. Gray and S. Wotherspoon, "Adaptive submodel selection in hybrid models," *Frontiers in Environmental Science*, vol. 3, Aug. 2015.

[88] K. B. Young, E. J. Wailes, G. L. Cramer, and N. T. Khiem, *Vietnam's Rice Economy: Developments and Prospects*. Arkansas Agricultural Experiment Station, 2002.

[89] A. Smajgl, T. Q. Toan, D. K. Nhan, J. Ward, N. H. Trung, L. Q. Tri, V. P. D. Tri, and P. T. Vu, "Responding to rising sea levels in the Mekong Delta," *Nature Climate Change*, vol. 5, pp. 167–174, Feb. 2015.

[90] VNA, "Article 24, Land on law 2003, Vietnamese National Assembly," 2003.

[91] MONRE, "Circular no. 19/2009/TT-BTNMT: Detailing the establishment, regulation and evaluation planning, land-use planning, ministry of natural resources and environment of vietnam," 2009.

[92] VGSO, "Report of census for land-use area in 2000, Vietnamese General Statistics Office," 2000.

[93] MONRE, "Decision 1482 of Ministry of Natural Resources and Environment of Vietnam on publishing statistical data of land-use change in 2011," 2012.

[94] PCBT, "Report of land-use zoning in 2020 and the land-use plan for 5 early years 2011-2015 of Ben Tre province, People's Committee of Ben Tre province," 2011.

[95] H. Visser and T. de Nijs, "The Map Comparison Kit," *Environmental Modelling & Software*, vol. 21, pp. 346–358, Mar. 2006.

[96] J. Cohen, “A Coefficient of Agreement for Nominal Scales,” *Educational and Psychological Measurement*, vol. 20, pp. 37–46, Apr. 1960.

[97] E. Lambin and H. J. Geist, “Causes of land-use and land-cover change,” 2007.

[98] J. Wu, “Land Use Changes: Economic, Social, and Environmental Impacts,” *Choices*, vol. 23, no. 4, pp. 6–10, 2008.

[99] A. Drogoul, N. Q. Huynh, and Q. C. Truong, “Coupling Environmental, Social and Economic Models to Understand Land-Use Change Dynamics in the Mekong Delta,” *Frontiers in Environmental Science*, vol. 4, Mar. 2016.

[100] T. Chi Quang, *Integrating cognitive models of human decision-making in agent-based models : an application to land use planning under climate change in the Mekong River Delta*. phdthesis, Paris 6, Paris, France, French.

[101] Q. C. Truong, P. Taillandier, B. Gaudou, M. Q. Vo, T. H. Nguyen, and A. Drogoul, “Exploring agent architectures for farmer behavior in land-use change. A case study in coastal area of the Vietnamese Mekong Delta,” vol. MABS 2015 Workshop, May 5-8, 2015, 2015.

[102] Vo, Q. M. and Le, Q. T., “Soils of the Mekong Delta, based on WRB (World Reference Based) - FAO system, at the scale 1/250.000,” *Collection of the research scientific of College of Agriculture and Applied Biology, Can Tho University*, vol. 2006, pp. 147–156, 2006.

[103] Soil Resources, Management and Conservation Service, ed., *A framework for land evalution*. No. 32 in FAO soils bulletin, Rome: FAO, 2. print ed., 1981.

[104] M. T. H. Nguyen, M. Q. Vo, and G. T. Vo, “Suitable land use zoning in line with climate change scenarios for three coastal districts in Ben Tre province,” *Vietnam Soil Science*, vol. ISSN 0868 3743, 2014.

[105] G. G. Judge, R. C. Hill, W. Griffiths, H. Lutkepohl, and T.-C. Lee, “Introduction to the theory and practice of econometrics.,” 1988.

[106] L. Hamill and N. Gilbert, “Social circles: A simple structure for agent-based social network models,” *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 2, p. 3, 2009.

[107] VGSOBT, “Statistical yearbook of Ben Tre province 2010, Vietnamese General Statistics Office of Ben Tre province,” 2010.

[108] J. Ahnström, J. Höckert, H. L. Bergeå, C. A. Francis, P. Skelton, and L. Hallgren, “Farmers and nature conservation: What is known about attitudes, context factors and actions affecting conservation?,” *Renewable Agriculture and Food Systems*, vol. 24, p. 38, Mar. 2009.

[109] V. Bewick, L. Cheek, and J. Ball, “Statistics review 9: One-way analysis of variance,” *Critical Care*, vol. 8, no. 2, pp. 130–136, 2004.