2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)

Modelling Multi-Stakeholder Systems: A Case Study

Michel Oey, Zülküf Genç
Amineh Ghorbani, Huib Aldewereld
Frances Brazier
Faculty of Technology, Policy, and Management
Delft University of Technology
The Netherlands

Reyhan Aydoğan
Catholijn M. Jonker
Faculty of Electrical Engineering,
Mathematics, and Computer Science
Delft University of Technology
The Netherlands

Reinier Timmer
Niek Wijngaards
Thales Research & Technology Delft
Thales Nederland BV
The Netherlands

***Abstract*—A contemporary governance challenge for governments concerns the biogas domain: what incentives and policies can lead to a viable biogas economy? To support addressing this challenge, a prototype of a simulator is constructed in which horizontal governance is applied in a multi-stakeholder context. This paper reports on the modelling and knowledge acquisition that led to the development of that prototype. Rather than (re)inventing tooling, three available agent-based modelling approaches are combined: the MAIA meta-model, OperA and GENIUS; with AgentScape as the agent-based middleware for the realisation of the simulator. The resulting simulator has been validated by biogas experts from Alliander (NL-based energy network company), leading to confirmation that our combined approach was useful for the analysis of this multi-stakeholder domain.**

I. INTRODUCTION

The emerging domain of biogas production poses a number of challenges regarding governance of that domain: what policies should a government impose to foster a healthy biogas economy? Within the Netherlands, a number of promising small-scale experiments are currently conducted (e.g., [1]), yet a consensus on a national governance approach is lacking. The biogas domain is characterised by multiple stakeholders, including biogas producers (e.g., farmers and water-treatment facilities), gas distributors and consumers. These stakeholders typically are independent of each other and need to arrive at a market price for goods and services through negotiation. The government can influence the market prices through incentives (e.g., subsidies) and policies. An open challenge is: how to assess what the effect is of a specific type of governance on the biogas economy? This challenge is addressed in the NeGoM project[1], in which a prototype of a simulator is realised to study the effects of horizontal governance in the biogas domain.

For modelling the multiple stakeholders in the biogas domain, i.e., by considering it as a multi-actor system, quite a number of modelling approaches are available. Some very specific (e.g., [2], [3]), and useful within their context, others much more broader (e.g., INGENIAS [4], Gaia [5]). This paper examines the possibility of combining a number of existing complementary modelling frameworks in order to facilitate modelling multiple aspects of a complex multi-actor system. The challenge is to maintain semantic coherence among the modelling approaches. Can some part of the output of one modelling approach be used as a partial input for another modelling approach? What changes need to be made and how can consistency be maintained? What advantages can be achieved from combining modelling approaches?

Rather than inventing a new modelling approach, the approach taken in the NeGoM project was to combine existing modelling approaches (each specialised to model different aspects of a multi-stakeholder system) and to use this combined modelling approach to create a simulation tool to evaluate a horizontal governance structure for the biogas domain. The complementary modelling approaches used are the MAIA meta-model, OperA, and GENIUS, where the simulator is prototyped on AgentScape - an agent-based middleware.

First, Section II briefly sketches the multi-stakeholder biogas domain, including requirements on modelling approaches. Subsequently, Sections III to VI introduce the MAIA meta-model, OperA, GENIUS, and AgentScape. Section VII describes our combined modelling approach. Section VIII discusses our achieved results, and Section IX concludes the paper with some conclusions.

II. SIMULATING HORIZONTAL GOVERNANCE IN THE BIOGAS DOMAIN

In the wake of the liberalization of energy markets and transition to the use of more renewable energy sources, the concept of self-governance or *horizontal governance* is gaining prominence. Not only is distributed generation emerging as a credible alternative to central electricity production, it is also becoming increasingly possible for villagers, neighbours, farmers, and small businesses, to organise the delivery themselves and switch from dependence on network companies to proactive and coordinated self-provision.

The central research focus of the NeGoM project is to investigate the effects of different types of governance for new energy markets. This investigation was scoped to the focus to assess the impact of horizontal governance in the biogas

[1]The full name of the project is *New Governance Models for Next Generation Infrastructures* and is funded by the Next Generation Infrastructures Foundation and subsidised by Alliander.

978-1-4799-4143-8/14 $31.00 © 2014 IEEE
DOI 10.1109/WI-IAT.2014.195

IEEE computer society

domain through simulations. That is, individuals or groups of people can exercise control over oneself or themselves: the rule of a community by its members. To this end, a simulator has been designed.

Biogas can be produced by farmers and water-treatment facilities. Farmers can collect biogas from the breakdown of manure and water-treatment facilities from sewage in devices called *digesters*. However, before biogas can be used by consumers, it must be upgraded as it contains contaminations. To get a viable biogas network running, many aspects need to be taken into account, such as subsidies, distribution distance, infrastructure costs, natural gas prices, consumer behaviour, etc. The NeGoM project researched whether farmers and water-treatment facilities could cooperate and share the cost of the biogas infrastructure to create a profitable biogas network. A prototype simulator was designed and built to investigate this type of governance.

The NeGoM project has chosen to use agent-based simulations[2]. Within such simulations each entity/stakeholder in the biogas energy market is modelled as a separate entity (agent) with its own behaviour and goals. The end result is a dynamic model that is able to capture the emergent behaviour of all entities in the system. It is important that all entities are autonomous and have their own behaviour with their own decision processes and preferences. Within a horizontal governance model, interactions and collaborations between stakeholders play an important role. Therefore, within the NeGoM project, special attention has been given to negotiations between stakeholders. Decisions made by stakeholders are dependent on the negotiations they have among each other.

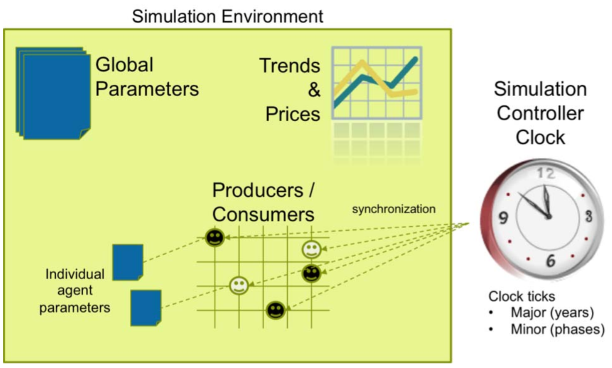

Fig. 1. Simulator overview

Figure 1 shows a very concise overview of the simulator. Each simulation run is determined by a scenario which sets a number of global parameters, individual parameters for agents (actors, such as preferences by stakeholders), trends and energy prices (the environment) and determines how many actors there are (e.g., producers, consumers). The simulation runs by simulating the actions of each actor over time, which is determined by a simulation controller clock. The simulated time period is scenario dependent and often encompasses 30 years. Within each year, multiple phases are distinguished within which actors can perform certain actions. The type of horizontal governance that is tested determines which actions can be performed.

Modelling horizontal governance in the biogas domain entails capturing knowledge on the following aspects:

- Multi-stakeholder: their roles, relations, circumstances, dependencies, and scenarios.
- Biogas domain: biogas production, cleaning, distribution, consumption, pricing, subsidies, etc.
- Organisational structures: organisations, roles, responsibilities and rules of stakeholders.
- Negotiation: knowledge and utility for multi-issue negotiations among stakeholders.

The MAIA, OperA, GENIUS and AgentScape modelling approaches each specialise on different aspects, and as such are complementary to each other. The main rationale for the selection of these four modelling approaches is the availability of expertise. The objective of the research is not to evaluate a single approach, but rather to investigate the combination of these approaches, without enforcing a full integration at the level of tooling. Each of these frameworks are briefly described in the following sections.

III. MAIA

MAIA (Modelling Agent systems based on Institutional Analysis) [7] is a meta-model that structures and conceptualises an agent-based model in a high level language. The concepts in the MAIA meta-model are a formalization of the Institutional Analysis and Development (IAD) framework of Elinor Ostrom [8], extended with concepts from other social science theories (Structuration [9], Social mechanisms [10] and Actor-centered institutionalism [11]).

MAIA has been designed to support the participatory development of agent-based simulations in order to bring this modelling approach within the reach of more researchers and practitioners, especially those who want to study the effect of policy instruments on behaviour at individual and aggregate level [7].

Furthermore, an online tool[3] supports the conceptualization process of agent-based models with MAIA. In this tool, the MAIA model (i.e., the conceptual model developed with MAIA) is observable and traceable through tables and diagrams and can therefore be used for communication with domain experts and problem owners for concept verification.

The MAIA meta-model views a socio-technical system as bounded in time and space, and shaped by social structure [9]. The structure of the system is both the means to organise the system as well as the outcome of that system [12]. It consists of many actors who perform actions and interact with each other in what is called an action arena. What happens in the action arena of the system leads to patterns of interaction and

[2]See [6] for an introduction

[3]See http://maia.tudelft.nl

outcomes that are judged on the basis of evaluative criteria which are defined by the analyst.

The MAIA meta-model is organised into five structures that serve as placeholders (i.e., categories) for related concepts. These will be explained in more detail next.

a) Collective Structure: The characteristics of the community or collective unit of interest are described in the collective structure. The collective structure defines agent types which represent individual or composite entities that make decisions, act and react in a social system.

Agents have *properties* (e.g., age and gender), *personal values* (e.g., social recognition, wealth), *physical assets* (e.g., cleaner) and *information* (e.g., investment costs). Agents take *roles* (e.g., producer) in the society to perform various actions. They have *intrinsic capabilities* such as eating and sleeping that are independent of the role they take in the society. The decision making procedure of agents for performing various actions is based on these attributes.

b) The Constitutional Structure: To be part of a social system (e.g., a biogas system), agents take *roles*, which places them in certain institutional settings. *Institutions* are sets of rules, norms and shared strategies that structure social behaviour and interaction [13]. Each role is created to serve an *objective* in the system. If an agent meets the condition to enact a role (e.g., live in a biogas neighborhood in order to be in the role of a consumer) certain responsibilities or capabilities become available or acceptable for him to perform. For example, the agent in the role of a biogas producer can search for collaboration with other biogas producers.

c) The Physical Structure: Individuals are also influenced by their physical surroundings. Physical components are the building blocks of the non-social environment that the agents are embedded in (e.g., digesters, cleaners). For example, farmers own a farm and consumers own houses. The type of these components is private because they belong to a person or a group of people. Physical components can also be shared among everyone in the system (*public*), such as a regulated grid. Each physical component may have *properties* (e.g., a digester has *capacity*). Physical components may also have *behaviours* (e.g., ageing) and *affordances* (i.e., what can be done with it; e.g., biogas can be cleaned).

d) The Operational Structure: The operational structure is viewed as an action arena where different situations take place, in which participants interact as they are affected by the environment and produce outcomes that in turn affect the environment. The agents, influenced by the social and physical setting of the system, perform actions in the action arena. The action arena contains all the *entity actions* that may execute during a simulation ordered by *plans*, which are in turn ordered by action *situations*. Entity actions have an *action body*, which is the actual activity the *performer* executes. Each entity action specifies the *preconditions* for the performer to perform an action (e.g., a consumer must have a demand for biogas before they can negotiate with producers) and the *updates* in the status of the system after the action is executed (e.g., unfulfilled demand of consumer = 0). Furthermore, the agent may have a decision-making criterion for performing an action (e.g., only produce biogas if it will lead to a financial profit), which may also be influenced by a related institution (e.g., a rule: the producer needs to pay a fine if he does not produce the contracted amount of biogas.).

e) The Evaluative Structure: Like any other software system, errors in simulations should be detected as early as possible starting from the analysis and conceptualization phase. The Evaluative Structure provides concepts with the help of which the modeller indicates what patterns of interaction, evaluation, and outcomes are of interest. The modeller identifies those variables that can serve as indicators for model validity (is it sufficiently realistic?) and model usability (will its implementation help to explore the question(s) to be addressed?).

IV. OperA & OperettA

The engineering of applications for complex and dynamic domains is an increasingly difficult process. Requirements and functionalities are not fixed a priori, components are not designed nor controlled by a common entity, and unplanned and unspecified changes may occur during runtime. There is a need for representing the regulating structures explicitly and independently from the acting components (or agents). Organization computational models, based on Organization Theory, have been advocated to specify such systems.

Organization models must enable the specification of global goals and requirements but cannot assume that participating actors will always act according to the needs and expectations of the system design. As such, organization models must support the specification of governance and interaction rules to guide participants' behaviours.

The OperA model [14] proposes a more expressive way for defining organizations by introducing an organizational model, a social model and an interaction model. This approach explicitly distinguishes between the organizational model and the agents that act in it. Agents become aware of the organizational rules via contracts that specify these rules. The agents are still fully autonomous in making decisions. OperA describes an operational organization in three parts: (1) the organizational model: roles, relations, interactions; (2) the social model: population of organization, linking agents to roles; and (3) the interaction model: describes interactions given organizational model and agents.

The organizational model contains the description of the roles, relations and interactions in the organization. It is constructed based on functional requirements of the organization. The social model and the interaction model are the link between the organizational description and the executing agents. Here the organizational rules are translated to contracts for the agents fulfilling the roles. OperA includes a formal language to describe those contracts.

In an operational organization the social model and the interaction model can be dynamic, because of agents entering or leaving the organization. The organizational model is in principal static as long as no structural changes are carried through. The administrative tasks to keep track of the different organizational models are specified in organizational roles.

Agents enacting roles in an organization are expected to have some minimal knowledge about the concepts that are used to set up social contracts. The contracts are described in deontic expressions. The agents need to know the deontic concepts permission, obligation and prohibition. Furthermore, the description includes relations between roles. The agent needs to know the meaning of such a relation. For example, a hierarchical relation between role r_1 and r_2 implies that a request from r_1 is interpreted as an obligation by r_2.

OperettA[4] [15] is an IDE (Integrated Development Environment) developed to support the design, analysis, and development of agent organizations using the OperA conceptual framework and methodology. It is intended to support software engineers and developers in both developing and documenting the various aspects of specifying and designing a multi-agent organization. OperettA enables the specification of organizational models. It provides separate editors for different components of organizational models; i.e., it has different (graphical) editors for each of the main components of an organizational model as defined in the OperA framework.

V. GENIUS

GENIUS, General Environment for Negotiation with Intelligent multi-purpose Usage Simulation, provides an open architecture to allow easy development and integration of negotiation agents. It can be used as a research tool on multi-issue negotiation [16], [17]. In short, it allows the use of existing negotiation strategies (from its repository), introduce new negotiation scenarios, elicit user preferences in terms of linear additive utility functions, design new negotiation strategies and algorithms, test negotiation strategies against state-of-the-art agents (designed by other researchers), and analyze the negotiation outcomes using different evaluation metrics. GENIUS is mainly focused on bilateral negotiation where two parties negotiate over a list of issues.

GENIUS includes three modules:

- *Negotiation scenarios*: A negotiation scenario consists of a negotiation domain describing the negotiation issues and at least two preference profiles defined on that domain. Basically, a domain is a list of issues where each issue has a set of possible values (e.g., discrete enumerated value sets or integer-value sets). For example, the possible values for holiday location might be `Barcelona`, `Rome`, `Istanbul`, and `Amsterdam`. New scenarios can be created and added to the scenario repository. To create a new scenario, first the negotiation domain must be defined and then preference profiles on this domain must be created.
- *Negotiating agents*: A negotiating agent implements the Agent Java API. Customised agents can be created based on a provided Agent skeleton, which requires certain methods such as *receiveMessage*, *init* and *chooseAction* to be specified. These customised agents can also be added to the repository.
- *Negotiation protocols*: A negotiation protocol governs the interaction between negotiating parties by determining how the parties interact/exchange information, and when a negotiation is terminated. For bilateral negotiation, GENIUS provides the Alternating Offers Protocol [18].

Via a graphical user interface, GENIUS supports the setup of a single negotiation or a tournament. Figure 2 shows a screenshot of GENIUS interface showing the results of a specific negotiation session. The negotiation log shows each action that the agents took during the negotiation as well as a summary of the negotiation results. The negotiation dynamic chart displays optimal solutions such as Pareto efficient frontier, Nash product, and Kalai-Smorodinsky solutions [18]. It also shows each agent's moves in the outcome space and any agreement reached. Consequently, researchers can evaluate how good the reached agreement is according to these metrics.

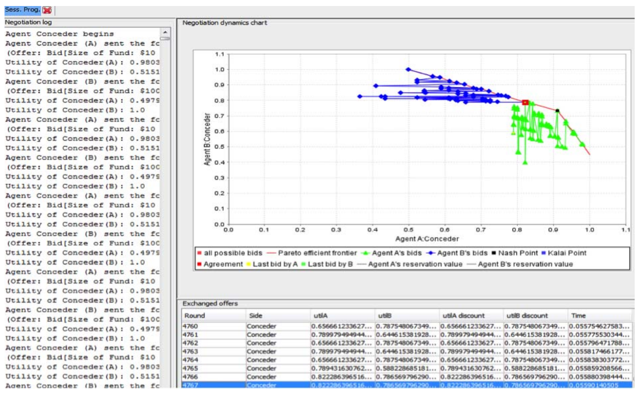

Fig. 2. GENIUS Interface Showing the Results of a Specific Negotiation Session

VI. AGENTSCAPE

AgentScape[5] [19] is a multi-agent platform that provides the middleware infrastructure needed to support mobility, security, fault tolerance, distributed resource and service management, and services access to agent applications. The multi-level AgentScape middleware infrastructure has been designed to be extensible.

Intelligent software agents are mobile applications that are launched by a user or another agent and obtain rights and permissions to use resources and access data. Agents contain algorithms that work towards fulfilling the user's goals and run as independent, asynchronous tasks. Agents have the ability to be created; to migrate between hosts; to communicate with other agents and their owner, and to access resources and services. Agents cannot function without a middleware system to provide the interface to allow them to move between sites in a distributed system. Typically this middleware provides a set of application programming interfaces (APIs) to allow agent

[4]See http://www.operettatool.nl

[5]See http://www.agentscape.org

application developers to easily access remote resources, such as web services on specific servers.

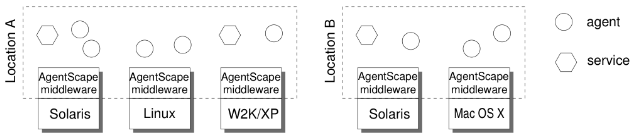

Fig. 3. Conceptual model of the AgentScape middleware environment (adapted from `http://www.agentscape.org`)

Within AgentScape, agents are active entities that reside within *locations*, and services are third-party software systems accessed by agents hosted by the AgentScape middleware (see Figure 3). Agents in AgentScape can communicate with other agents and can access services. Agents can migrate from one location to another.

The leading principle in the design of the AgentScape middleware has been to develop a minimal but sufficient open agent platform that can be extended to incorporate new functionality or adopt (new) standards into the platform. This design principle has resulted in a multi-layered architecture with (1) a small middleware kernel, called the AgentScape Operating System (AOS) kernel [20], that implements basic mechanisms and (2) high-level middleware services that implement agent platform specific functionality and policies. The current set of middleware services includes agent servers, host managers, location managers, a lookup service and a web service gateway. The policies and mechanisms of the location and host manager infrastructure are based on negotiation and service level agreements [21].

VII. Combining Modelling Approaches

The three modelling approaches MAIA, OperA and GENIUS are combined as shown in Figure 4, where results 'produced' by a modelling approach are used as a starting point by another modelling approach. Although the figure shows a directed flow, the process used during the NeGoM project can be characterised as *agile*. Given the progress of the modelling of the biogas domain, manual (paper-based) or prototype simulations were used to engage in multiple iterations, including validation by the biogas experts from Alliander. These iterations helped to focus modelling effort on those aspects and details that were relevant to the realisation of the simulator.

Figure 4 also shows how the different modelling approaches are related in the project. MAIA is used to conceptualise the biogas system itself, including its actors and their actions, and document domain knowledge. For example, MAIA is used to model the farmers, the water-treatment facilities, the consumers (such as, small neighbourhoods, small and large businesses), the physical infrastructure components, the process of creating biogas from manure, the goals of each actor, physical limitations, etc. MAIA is also used to model environment factors such as natural gas price fluctuations.

The OperA framework is then used to add organization elements, such as coordination mechanisms, organizational objectives, governance models, abstract protocols, and abstract negotiation patterns between biogas producers and consumers.

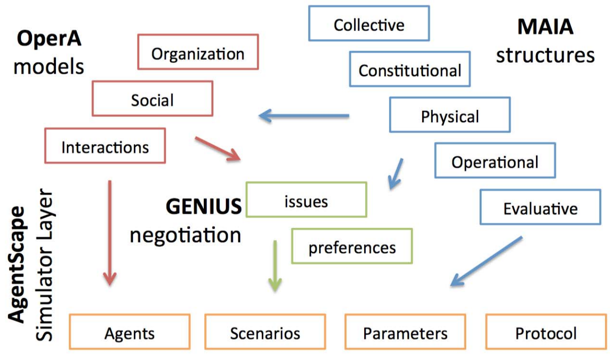

Fig. 4. Combination of modelling approaches.

In principle, MAIA and OperA could be used to model most of the biogas domain independently, but each modelling framework has its own focus. MAIA is able to capture the actors and their actions in the domain (dynamics), and OperA is able to capture the coordination and organizational aspects of the domain. The combination of the two provides a more complete model of the biogas system for simulation.

The negotiations that the stakeholders can perform are already modelled in the MAIA and OperA frameworks, but only on an abstract level. The actors and objects involved in negotiations have been modelled in MAIA, while the interaction model of the stakeholders is modelled in OperA. The outcome of this is used by GENIUS to perform the actual negotiations in the simulations.

Together with the scenarios, the MAIA and OperA models are fed into the simulator which runs on AgentScape and uses GENIUS to deal with negotiations between stakeholders during the simulations. The simulator itself uses agents to represent the actors inside the simulations, as well as other necessary entities (e.g., a bank, natural gas price issuer, etc.).

The governance model created in OperettA defines the organization model of the involved entities with roles, actions, objectives and norms. This organization model is mapped to software agents that run in the simulator on the distributed AgentScape platform. Each agent corresponds to a role and executes the actions of that role based on the objectives and norms belonging to that role.

For the realisation of the simulator, some of the tooling of the modelling approaches has been provisionally integrated with AgentScape, which plays the role of 'integrating' middleware. AgentScape is enriched by a small additional layer that provides the required simulation capabilities. The information and knowledge specified by MAIA is (for now) manually encoded in the agents and negotiation strategies. The simulator is populated by means of a scenario description and an accompanying OperettA configuration file. GENIUS provides a 'negotiation-service' (that internally uses GENIUS-negotiation-agents) that can be employed by the (producer & consumer) agents that are configured by OperettA.

The integration of MAIA, OperettA and AgentScape contains two parts. In the first part, the characteristics of the organization model in OperA are transferred into the logic of software agents through XML configuration parameters. In the second part, a version of a software action library is developed that implements available actions for agents, as specified in the MAIA model.

For example, in a (simplified) biogas domain three roles can be identified: agricultural farms, water-treatment facilities, and consumers. The former two are biogas producers, the latter is a biogas consumer. Furthermore, each role can perform different *actions*, such as invest, negotiate, collaborate, buy, etc. Different horizontal governance types can be implemented by changing when and how biogas producers can collaborate and changing when and on what producers and consumers can negotiate. All this knowledge is captured in the MAIA and OperA models.

Next, for the simulation, these models are transferred into the logic of executable agents within the AgentScape middleware. For this purpose, an *action library* and a *generic agent* have been developed. The action library contains executable implementations of the actions defined in MAIA. The OperA models specify when and which of these actions are called by which roles. The generic agent is a basic executable simulation agent and can be configured to call actions from the action library. During the setup of the simulation, the XML output of the MAIA and OperA models are used to configure the generic simulation agents, thereby implementing agents that perform the roles as specified in the modelled domain. This instantiation process can easily be automated and results in executable simulation agents (i.e., producer and consumer agents) that can run within the AgentScape middleware.

These producer and consumer agents use GENIUS to mediate negotiations on contracts for biogas delivery between them. The GENIUS library is integrated into AgentScape via AgentScape's service interface, which is a mechanism to make the functionality of new (external) libraries and services accessible to agents. GENIUS can be configured through parameters which determine what the negotiations are about or influence the negotiations themselves. Which parameters can be set have been identified and specified in the domain model using the modelling tools. However, the actual values of the parameters are chosen at runtime by the agents for each negotiation separately and depend on the current state of the simulation. Examples of such negotiation parameters are min/max prices, amount of gas, contract duration, and negotiation strategies.

Figure 5 depicts how the simulator can be used for running different scenarios. Each scenario describes the characteristics of consumers and producers in a specific energy domain, as well as other relevant parameters, including negotiation parameters, willingness to collaborate, and price fluctuations for energy. As can be seen in the figure, the MAIA and OperA models are used during the design of the simulation model. They capture the knowledge of the domain itself and

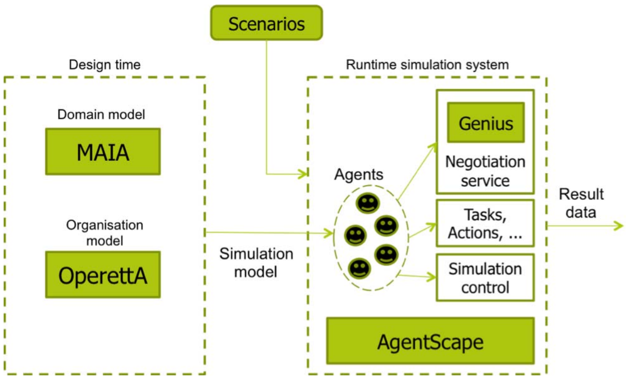

Fig. 5. The integrated simulation environment.

the actions that roles take. GENIUS and AgentScape are used during the simulation runs themselves and operationalise the simulation model by performing the actions and negotiations defined in the MAIA and OperA models within specified scenarios.

VIII. DISCUSSION

The work presented in this paper entails combining three agent-based modelling approaches: MAIA, Opera and GENIUS, and operationalising an agent-based simulator using the AgentScape agent-based middleware. The choice for these agent-based approaches was based on availability of expertise, and the choice for the multi-stakeholder use-case was dictated by the project's domain owner Alliander, an energy network company in the Netherlands (our 'end user').

In the project, the simulator was used to evaluate horizontal governance in the biogas domain where producers and consumers could negotiate about the delivery of biogas using contracts. Furthermore, producers could collaborate in order to share the cost of the biogas infrastructure. The simulator allowed evaluating the effects of governance in different scenario's (fluctuating natural gas prices, government subsidies, etc.) and determining under which conditions a stable and sustainable configuration emerged. Using different existing modelling approaches allowed the project to quickly model the domain and add the necessary details on, for example, domain knowledge, organizational structures, and negotiation patterns and translate them into a running agent-based simulation model.

Different approaches exist that could have been used in the project as well. For example, Gaia [5] is a methodology for agent-oriented analysis and design. Gaia can be used to model systems from requirements to a detailed design ready for implementation. Its conceptual framework distinguishes two models: a roles model and an interactions model. The former contains all the roles and their responsibilities and permissions. The latter contains the interactions (protocols) between roles. In the design phase, these models are elaborated in an Agent, a Services, and an Acquaintance Model. However, Gaia does not support norms.

INGENIAS [4] is an agent-based software engineering methodology that is able to design multi-agent systems. It provides a comprehensive meta-model that can conceptualise software in self-defined graphical notations and diagrams. Organisational aspects of a system are modelled by defining roles, groups, and organizations in different viewpoints. INGENIAS also supports code generation. It explicitly addresses different aspects of simulations, such as scheduling, and provides guidelines on how to build a simulation from INGENIAS models [22]. However, the social aspects of agents are not fully addressed in INGENIAS. There is no support for norms and regulations. INGENIAS defines roles but with a different meaning to organizational roles.

The remainder of this section presents some lessons learned from the project's effort. Briefly summarised, the main findings are:

- Maintaining coherence: Maintaining coherence while (re)modelling and developing the prototype of the simulator was a major challenge. The MAIA model provided a sound base on which the other modelling frameworks could build on. However, it required effort to maintain the semantic coherence between the models. In the project, a social construct (namely: team-formation) among all the participants was used to assure coherence across modelling approaches.
- Expertise availability: Without the availability of experts on each of the modelling approaches, it would have been impossible to achieve the current level of integration.
- Willingness to learn: Each of the team-members needed to be willing to learn about other modelling approaches, by interaction with respective experts.
- Iterations: We exploited the subsequent additions of details when applying the modelling frameworks in this order: MAIA, OperA/OperettA, GENIUS, AgentScape. These details were used as a means to further investigate the models of the domains at higher levels of abstraction. In general, this achieved multiple 'yo-yo' down & up iterations, until the prototype of the simulator yielded results that could be validated by biogas domain experts.
- Validate early: Even when insufficient details were available to develop a working prototype, paper-based examples and incomplete prototypes were used to validate our modelling progress with biogas domain experts, while also ensuring that we often engaged in combining our modelling approaches, thereby avoiding 'late' integration and running the risk of being unable to deliver our results.

At this moment, there is insufficient evidence to make statements about generalised applicability of these combined modelling frameworks. The current results are promising, and the work will be continued on a use-case by use-case basis, to further deepen the understanding of modelling multi-actor, socio-technical systems.

IX. CONCLUSIONS

This paper describes results attained in the NeGoM project on combining three modelling approaches, MAIA, OperA and GENIUS and operationalising the results with an agent-platform AgentScape to model the multi-stakeholder biogas domain and develop a prototype of a simulator to investigate the viability of horizontal governance in the biogas domain. The resulting simulator has been validated by biogas domain experts from Alliander, which provides substance to our findings that the combination of the modelling approaches was fruitful. Further insights on the combination of these modelling approaches have to be gathered by application to additional use-cases.

ACKNOWLEDGMENT

The work reported on has been conducted in the *New Governance Models for Next Generation Infrastructures* (NeGoM) project (2011-2013) and is funded by the Next Generation Infrastructures Foundation[6] under project number 09.14. and subsidised by Alliander[7]. The authors are grateful to the (biogas) experts available at Alliander for their cooperation with the modelling and knowledge acquisition.

REFERENCES

[1] Bionet: mix van biogas en aardgas voor woonwijk. retrieved from http://www.agentschapnl.nl/content/mix-van-bio-en-aardgas-voor-woonwijk, 2013.

[2] R. Bagni, R. Berchi, and P. Cariello, "A comparison of simulation models applied to epidemics," *Journal of Artificial Societies and Social Simulation*, vol. 5, no. 3, 2002.

[3] F. Bousquet, I. Bakam, H. Proton, and C. Le Page, "Cormas: Common-pool resources and multi-agent systems," in *Tasks and Methods in Applied Artificial Intelligence*. Springer, 1998, pp. 826–837.

[4] J. Pavón, J. J. Gómez-Sanz, and R. Fuentes, "The ingenias methodology and tools," *Agent-oriented methodologies*, vol. 9, pp. 236–276, 2005.

[5] M. Wooldridge, N. R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 285–312, Sep. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1010071910869

[6] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the 37th Conference on Winter Simulation*, ser. WSC '05. Winter Simulation Conference, 2005, pp. 2–15. [Online]. Available: http://dl.acm.org/citation.cfm?id=1162708.1162712

[7] A. Ghorbani, P. Bots, V. Dignum, and G. Dijkema, "MAIA: A framework for developing agent-based social simulations," *Journal of Artificial Societies and Social Simulation*, vol. 16, no. 2, 2013.

[8] E. Ostrom, *Understanding Institutional Diversity*. Princeton University Press, 2005.

[9] A. Giddens, *The constitution of society: Outline of the theory of structuration*. Univ of California, 1984.

[10] P. Hedström and R. Swedberg, "Social mechanisms," *Acta Sociologica*, vol. 39, no. 3, pp. 281–308, 1996.

[11] F. Scharpf, *Games real actors play: Actor-centered institutionalism in policy research*. Westview Press, 1997.

[12] A. Giddens and J. Turner, *Social theory today*. Stanford Univ Pr, 1988.

[13] E. Ostrom, R. Gardner, and J. Walker, *Rules, games, and common-pool resources*. University of Michigan Press, 1994.

[14] V. Dignum, "A model for organizational interaction: based on agents, founded in logic," Ph.D. dissertation, University of Utrecht, 2004.

[15] H. Aldewereld and V. Dignum, "Operetta: Organization-oriented development environment," in *Languages, Methodologies, and Development Tools for Multi-Agent Systems*. Springer, 2011, pp. 1–18.

[6] See http://www.nextgenerationinfrastructures.eu

[7] See http://www.alliander.nl

[16] K. Hindriks, C. M. Jonker, S. Kraus, R. Lin, and D. Tykhonov, "Genius: Negotiation environment for heterogeneous agents," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, ser. AAMAS '09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 1397–1398. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558109.1558313

[17] R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, and C. M. Jonker, "Genius: An integrated environment for supporting the design of generic automated negotiators," *Computational Intelligence*, vol. 30, no. 1, pp. 48–70, 2014.

[18] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.

[19] B. J. Overeinder and F. M. T. Brazier, "Scalable middleware environment for agent-based internet applications," in *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, ser. Lecture Notes in Computer Science, vol. 3732. Copenhagen, Denmark: Springer, June 2004, pp. 675–679.

[20] G. van 't Noordende, F. M. T. Brazier, and A. S. Tanenbaum, "A security framework for a mobile agent system," in *Proceedings of the 2nd International Workshop on Security in Mobile Multiagent Systems (SEMAS 2002), associated with AAMAS-2002, Bologna, Italy*, ser. DFKI Research Report RR-02-03, K. Fischer and D. Hutter, Eds. Deutsches Forschungszentrum fur Kunstliche Intelligenz, http://www.dfki.de, July 2002, pp. 43–50.

[21] D. G. A. Mobach, B. J. Overeinder, and F. M. T. Brazier, "A ws - agreement based resource negotiation framework for mobile agents," *Scalable Computing: Practice and Experience*, vol. 7, no. 1, pp. 23–36, 2006.

[22] C. Sansores and J. Pavón, "Agent-based simulation replication: A model driven architecture approach," *MICAI 2005: Advances in Artificial Intelligence*, pp. 244–253, 2005.