World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:11, 2012

AGENTMAP: A Conceptual Meta-Model of Interacting Simulations

Thomas M. Prinz Wilhelm R. Rossak and Kai Gebhardt

Abstract—A straightforward and intuitive combination of single simulations into an aggregated master-simulation is not trivial. There are lots of problems, which trigger-specific difficulties during the modeling and execution of such a simulation.

In this paper we identify these problems and aim to solve them by mapping the task to the field of multi agent systems. The solution is a new meta-model named *AGENTMAP*, which is able to mitigate most of the problems and to support intuitive modeling at the same time. This meta-model will be introduced and explained on basis of an example from the e-commerce domain.

Keywords—Multi Agent System, Agent-based Simulation, Distributed Systems, Meta-models.

I. Introduction

IN the context of the *SimProgno* research project[1] funded by the German Federal Ministry of Education and Research an integrative framework for the multi-dimensional modeling, simulation and forecast of e-commerce functional chains was needed. In doing so, existing different simulations, named *sub simulations*, had to be aggregated within the e-commerce domain to achieve the targeted and aggregated master-simulation.

E-commerce companies practice business in the internet. This contains nearly all processes of the ordinary business trade. For example, a typical process in e-commerce is the visitor-sided cycle of buying in an online shop (see figure 1). In the process every *internet user* may visit a *homepage* (the online shop) and take a look at the products offered. If the *visitor* likes a product, then he puts it in his personal, virtual shopping cart. After taking the decision to buy the products in his shopping cart, the visitor becomes a *customer* and his personal profile including his *address* and the *total sum of payments for the shopping cart* will be weighted in the so called *scoring* process. Based on the score of the weighting process the online shop decides which *payment methods* will be offered to the customer. At the end, the customer chooses one payment method. Now he can leave the online shop or he takes a look at other products offered. The visitor, respectively customer, is able to leave the online shop at every time. The process, as discussed here, is based on the example in [1].

The consequences and side effects when changing such a process (e.g. the addition of new products, or payment methods, or something else) are not trivial. Therefore, it is essential to *simulate* the process and its modifications, simply to be able to understand all consequences and side-effects. In general, such a *simulation* is a replication of the behavior and the structure of the real-world system into a computer model [2].

To reduce the complexity of the simulation and to leverage on existing partial solutions, we will subdivide the (master-)simulation into sub processes. One sub process will then be simulated as a stand-alone simulation and embodies, in dependency to the other sub processes, (see figure 2) one step of the integrated model. This makes it, however, necessary to create a conceptual model which supports the guided combination of individual, non-aggregated simulations.

Processes in the e-commerce domain (and so their simulation) are not free of social aspects, because it is the customer who decides on the success or flop of an online shop. Thus, *agent-based* simulations fit this type of simulation best, simply by building on the agent paradigm with its highly autonomous entities and their emerging interaction patterns [3]. This argument is supported by the fact that the agent paradigm is not only a computer science research area, but also well established in sociology, biology, and ecology [4].

As agents are a fairly young research domain and the variety of application areas they have entered is impressive,

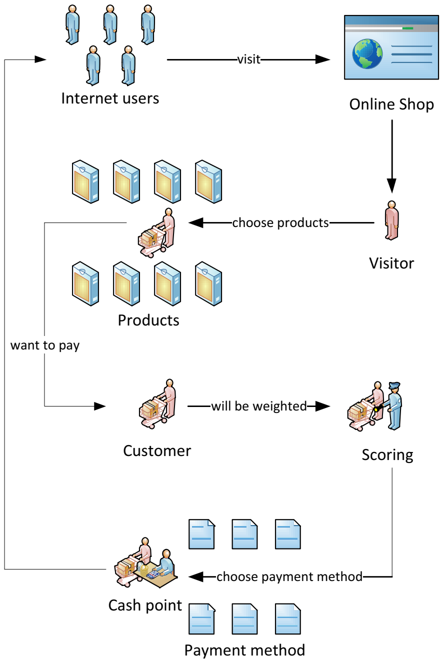

Fig. 1: Schematically presentation of the visitor-sided cycle of buying in an online shop

Th. Prinz, W. Rossak and K. Gebhardt are with the Chair of Software Engineering, Department of Mathematics and Computer Science, Friedrich Schiller University, Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{*Thomas.Prinz, Wilhelm.Rossak, Kai.Gebhardt*}*@uni-jena.de*

[1] Support code: 01IS10042B

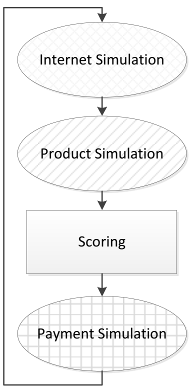

Fig. 2: Sub-division of the process in sub processes

there is not a general and consistent definition of agents and their environment [5]. This holds especially for agent-based simulations. Even though the literature contains many definitions (for example [6]–[11]), none of them is conclusive in the context of simulations. We call this the *Definition Problem*.

Furthermore there is a multiplicity of communication languages in different multi agent systems and, as a result, also in agent-based simulations, as explained in [12]. *KQML* [13] of Labrou and Finin and *FIPA ACL* [14] of the *Foundation for Intelligent Physical Agents* are prominent examples, but agent communication is also possible with a light-weight message based protocol [15]. This problem of heterogeneity of communication languages causes trouble during the combination of two individual, agent-based simulations [16]. Also explained in [16] the inhomogeneity of the implementation languages and agent architectures is an additional problem to the reusability of agent systems.

In the context of this paper the problem of heterogeneity in communication and implementation languages will be named *Language Problem*.

Because of the *Language Problem* the communication between different multi agent systems remains undefined. So it is difficult to determine which agents are able to communicate with each other, to handle restrictions, and to structure the interactions between agents to reduce the complexity of the resulting system. In the context of this paper, the difficulty to restrict interactions in multi agent systems in a useful way will be named *Restriction Problem*.

The complexity of combined multi agent systems without a solution of the *Restriction Problem* would result in a high work load to handle the system. A theoretical solution would be the reimplementation of all involved multi agent systems, based on a unified paradigm. However, this is impractical and, in addition, complexity would grow further if one of the involved subsystems has to accomodate to a necessary change. So the concept has to be redesigned from scratch. The purpose is to minimize the complexity involved in the combination of multiple agent systems. We call this the *Complexity Problem*.

If agent systems are to be combined directly, there would have to be a specific mapping between the input and output parameters of all involved systems. In addition, as opposed to workflow languages, the inputs and outputs of multi agent systems are produced on a continuous basis, not in discrete steps. Thus, we suggest handling the mapping during the combination of the systems, so to speak in advance. In this paper the corresponding difficulty will be named *Integration Problem*.

To integrate systems, it is a common approach to describe them on an abstract layer (a meta-model). It is then possible to generate tools which support the automatic and easy integration of all sub systems. In the best case, an instance of the model will be able to be executed or transformed directly to code, while the meta-level remains to be geared to human cognitive understanding.

The *Agent/Role/Group* model of Ferber and Gutknecht (see [16]–[18]) as well as the model of dynamic role assignment by Odell et al. [19] were chosen by our team as the starting point for such an abstract (meta-)model. They form the basis for our concept of a new meta-model named *AGENTMAP*. This will be presented in the next section II, forming the core of our paper. Section III shows the application of the *AGENTMAP* model to simulations. This will be followed by the state of the art in section IV. We finish with a conclusion and a short outlook (section V).

II. The AGENTMAP Meta-Model

The *AGENTMAP* meta-model is based on the *Advanced Agent/Group/Role* meta-model defined in the technical report [20] in more detail.

The *AGENTMAP* meta-model consists of four major concepts, the *agent*, the *group*, the *role* and the *agent map*. The interrelation of these concepts is illustrated in figure 3. Every concept abstracts another aspect of a scenario and helps to model intuitively. The aspect of active and interacting objects in a scenario will be implemented with the concept of the *agent*.

Definition 1 (Agent): An agent *in the* AGENTMAP *model is an active entity that is only able to communicate [18]. It plays* roles *and enters* groups*. An agent is always assigned to a group and plays at least one role available in this group.*

According to that, agents are all objects in a scenario and can interact with other objects. Interacting entities of the shopping process of section I are *humans* and the *online shop*. All of them will be represented by agents. As representative of a human, an agent plays the role of an *internet user*, *visitor* or *customer*. On the other hand the online shop consists of multiply agents, which represent *products*, *sellers* and the *scoring process*. They are all to be bound to special groups. In contrast, a human representative agent can change from the internet into the online shop. It has also the capability to propagate through the online shop as indicated in figure 1. A part of this process is illustrated in figure 4. Transformed into the *AGENTMAP* model, the result is as shown in figure 5. Besides the agents one can see the *internet* and *online shop*

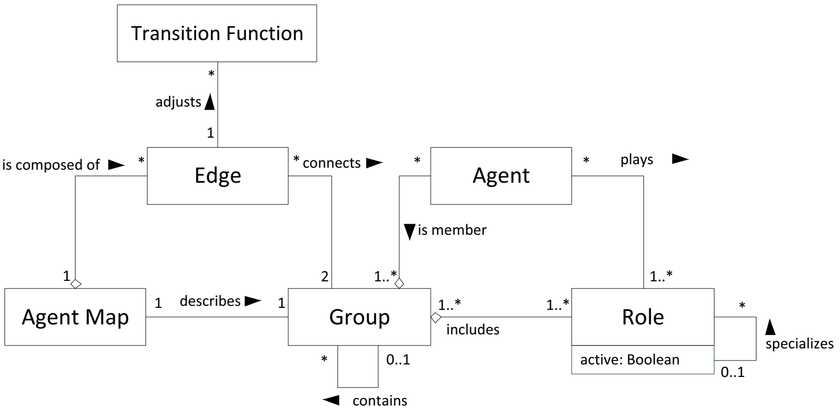

Fig. 3: The conceptual AGENTMAP model

groups. If an agent enters the *online shop* group, it will play the role of a visitor. The groups are the second major concept of the *AGENTMAP* model.

Definition 2 (Group): A group *is a set of agents [18] and a non-empty set of roles [21]. Groups are not* disjoint, *that means, that agents and roles could be associated also to other groups. If an agent is part of a group, then it has to play at least one of the roles available in the group.*

In addition a group may consist of sub groups, *which are ordinary* groups. *The case, that one group is a sub group of itself is ruled out. A* sub group *features* exactly *one* parent group. *A group without a sub group will be named* atomic.

A group describes a *subject-specific* aggregation/decomposition of agents to disassociate them from other agents. A possible subject-specific decomposition of the shopping process is to separate it into the *internet*, an *online shop*, a *product choice*, a *scoring* and a *payment process*. An online shop is a part of the internet and the product choice, the scoring and the payment process are parts of the online shop. So the sub groups of the online shop are closed and *atomic*. Each group owns subject-specific roles. According to that the *visitor* and *product* role will be associated to the product choice, the *customer* and *seller* role to the payment process, the *scoring* role to the scoring, the *visitor* role to the online shop and finally the *internet user* role to the internet. The internet and online shop groups are illustrated with their roles in figure 6 and extend figure 5. Roles are visualized by associated boxes. The role is the third major concept of the *AGENTMAP* model.

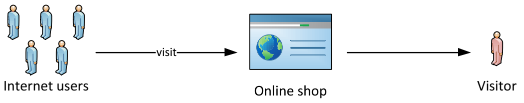

Fig. 4: Schematically presentation of a part of the shopping process in the online shop

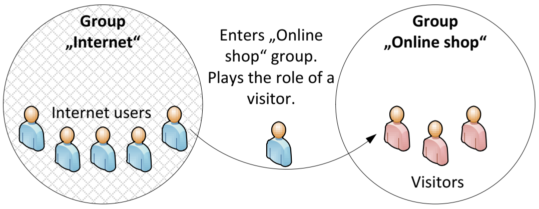

Fig. 5: Presentation of the part of the shopping process with the AGENTMAP model

Definition 3 (Role): A role is an abstract representation of the behavior, the service or the identification of an agent inside one or more groups [18]. An agent can play multiple roles. A role that is associated with an agent could be assumed to be active *or* passive *and is composed of a set of states (for example the knowledge or the properties of an agent). If an agent plays more than one role (active or passive), there can be (identical) states which are associated with more than one role. In such a case a state change in the one role may also trigger a state change in the other role [22]. Furthermore, a role can be a specialization of another, more general role. In this case a hierarchy of roles is established.*

So a role serves as description of an agent's behavior and

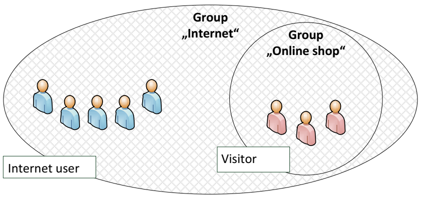

Fig. 6: Illustration of a part of the shopping process via groups

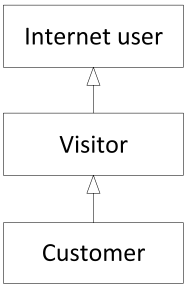

Fig. 7: Roles of the shopping process, which belong together

knowledge in a group. The subject-specific association of the roles to the groups of the shopping process was explained. What is missing is a closer inspection of the possible cohesion of roles. From our perspective, two roles belong together iff one role is a specialization of the other one. The roles of the shopping process, which belong together, are *internet user*, *visitor* and *customer*. This results in the fact, that a customer of an online shop is also a visitor and a visitor is also an internet user. So the customer role is a specialization of the visitor role and in turn the visitor role is a specialization of the internet user role. That is illustrated in figure 7.

A. Role change

Within the modeled shopping process, roles can be adopted, abandoned, or swapped by a human representing agent. The adoption, abandoning and swapping of a role is an elementary task of an agent in the *AGENTMAP* model. By applying these actions, it can accommodate itself to the group it wants to interact with. The interaction will be only practiced with *active* roles. Hence, it is possible that an agent plays its role *active* or *passive*. The currently adopted roles of an agent $agent$ are defined by the set $agent.roles$ and the current active roles are defined by the set $agent.active$. Its passive roles could be determined by the *relative complement* of the both sets ($agent.roles \setminus agent.active$). If an agent adopts, abandons or swaps a role, then it influences its current set of roles. The operations $agent.adopt(Role\ R)$, $agent.abandon(Role\ R)$ and $agent.swap(Role\ R_{in}, Role\ R_{out})$ perform role changes relating to the available set of roles. In analogy there exist two operations for the active or passive playing of a role: $agent.assumeActive(Role\ R)$ and $agent.assumePassive(Role\ R)$.

The operations involved and the need for role changes can be illustrated by inspecting the shopping process. The actions of an internet user in the process equate to those of figure 1 and are illustrated in figure 8 (highlighted text marks active roles).

B. Agent map

The interplay of roles, groups and agents defines the last major concept of the *AGENTMAP* model - the *agent map*. It models the aspect of a workflow within a scenario as an abstract concept.

Definition 4 (Agent Map): An agent map connects sub groups *within a* parent group *with directed edges. An edge allows for the* group change *of an agent from the starting sub group to the ending sub group. The requirements and limitations of such a group change are specified by* transition functions*. A transition function is defined following an available edge. If a transition function is satisfied, then a group change is possible. If an agent changes into a* non-atomic sub group*, this group provides an agent map. Thereby the agent changes automatically into one of the* start groups *of the non-atomic sub group. A start group is a sub group of a parent group. In addition to start groups, a parent group also owns* end groups*. An agent can only leave the parent group from one of the end groups, if there is an edge between the parent group and another group with a valid associated transition function.*

As a consequence, an agent map defines the interaction between agents of different groups. On the one hand, the interaction between agents of different groups is reduced to a group change. On the other hand, a group change is only possible between groups which are connected with an edge. Thus, the edge reflects the subject-specific workflow of a scenario. The description of the operational sequence of the example scenario defines the possible interaction sequence of an agent. An agent as *visitor* within the group *online shop* stands at the beginning in the atomic group *product choice*. So this group will be seen as *start group* of the agent map of the online shop (illustrated in figure 9). Inside the agent map of the online shop, the agent, as visitor, is able to change to the *payment process* group. There it can pay its products as a *customer*. Please note that there is a directed edge between the product choice and payment process groups. A customer can leave the payment process back to the product choice group, which causes a back edge. Also, leaving the online shop as visitor or customer is possible at any time. Hence, the product choice and payment process groups are end groups of the online shop. The scoring group of the online shop can only be reached by a seller agent from the payment process. It uses it to weight the customer. The weight gets the seller and returns to the payment process. So there exist two edges between those groups.

If there is an edge between two groups, then it is necessary to define at least one transition function.

Definition 5 (Transition Function): A transition function is a concept, which enables or disables the transition of an agent in relation to an edge. The nature of a transition function can be individual, but should consist of a set of rules *and optionally a* set of role changes *and a* set of arbitrary instructions.

Because of the fact, that a transition function can include every kind of instruction, it serves e.g. to create agents or to

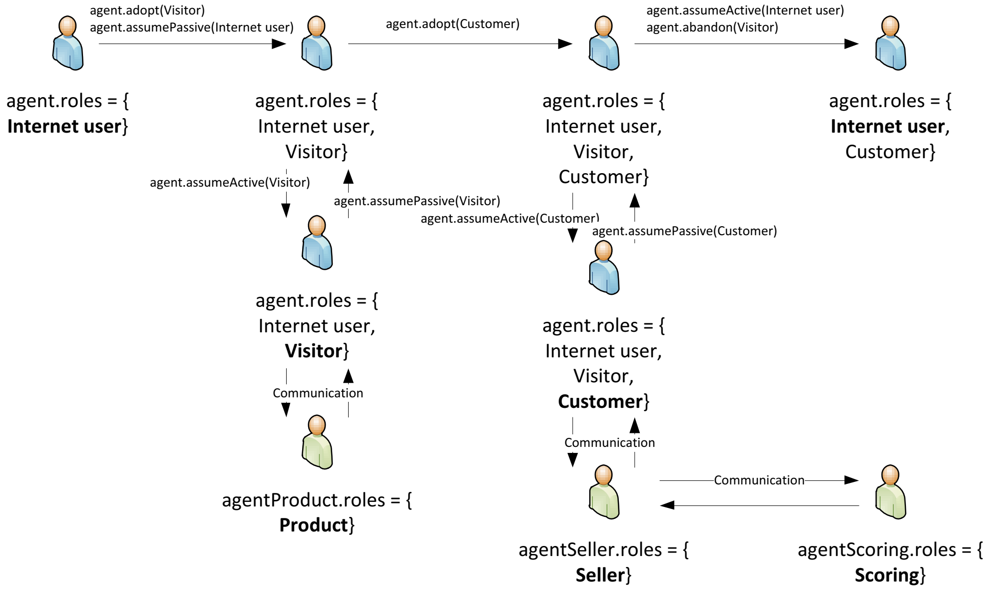
agent.adopt(Visitor)
agent.assumePassive(Internet user)
agent.adopt(Customer)
agent.assumeActive(Internet user)
agent.abandon(Visitor)
agent.roles = { Internet user}
agent.roles = { Internet user, Visitor}
agent.roles = { Internet user, Visitor, Customer}
agent.roles = { Internet user, Customer}
agent.assumeActive(Visitor)
agent.assumePassive(Visitor)
agent.assumeActive(Customer)
agent.assumePassive(Customer)
agent.roles = { Internet user, Visitor}
agent.roles = { Internet user, Visitor, Customer}
Communication
Communication
Communication
agentProduct.roles = { Product}
agentSeller.roles = { Seller}
agentScoring.roles = { Scoring}

Fig. 8: The use of the AGENTMAP model for the description of operational sequence of the shopping process

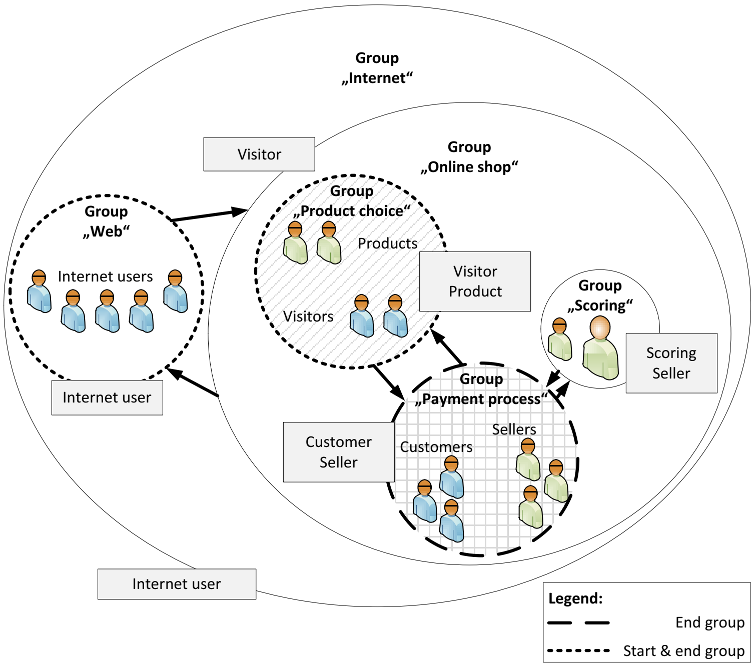

Fig. 9: The complete shopping process implemented as instance of the AGENTMAP model

execute role changes. Thus, there are defined moments for role changes.

Our understanding of the transition function will be illustrated by taking a look at the example of the shopping process. For optic reasons the focus lies on the transition function of the edge between the product choice and the payment process. For this edge, there exists a set of rules:

1) The payment process could only be started, if the shopping cart is filled.
2) The payment process could only be carried out with a customer.
3) Only a visitor can be a customer.
4) Every customer is associated to a seller.

These rules will be implemented with transition functions in the *AGENTMAP* model. A representation of an implementation in *BPMN 2.0* is illustrated in figure 10. The transition functions of the other edges are explained in [20].

Transition functions are powerful tools to control the interactions within combined multi agent systems. It is now possible to conditionally guard the interaction between agents of different groups, providing for a set of stringent rules or leaving this process completely unguarded.

C. Conclusion

These restrictions to the interactions within a combined multi agent systems reflect the introduced *Restriction Problem* of section I and resolve this issue. Thereby the understanding of interactions becomes relatively simple and the model wins on acceptability. Associated cost during elaboration and training are reduced and influence the *Complexity Problem* positively. Because of the improved overview, group and role hierarchies also support traceability and reusability of such a model positively. It is now possible to reuse a group of one model in another model. It is also possible to replace a group with another group with the same properties (roles and transition functions). Altogether the *AGENTMAP* model now wins on understandability. We build on this property also during the modeling of complete scenarios, improving intuitive understanding. A good example is the fairly simple translation of the example scenario in an instance of the *AGENTMAP* model. Thus, this influences the *Complexity Problem* positively and leads us to the assumption that we could more or less solve it based on the presented mechanisms.

Likewise, the arrangement of interfaces with the *AGENTMAP* model becomes simple, because a group is defined exactly with its transition functions and roles. This solves the hard core of the *Integration Problem*.

Unsolved remain the *Language* and *Definition Problem*. However, we regard them now as rather technical problems related to the instantiation of a meta-model and its specific implementation.

III. Aggregated simulations with the AGENTMAP Meta-Model

The intention of the *AGENTMAP* model was to build a conceptual model to combine and reuse sub simulations. By mapping the simulations to multi agent systems, we have found the resulting meta-model. This solves or handles most of the identified problems satisfactorily.

An single instance of the model is understandable and intuitive. However, we have not yet discussed its usability during the combination of multiple simulations. Up to this moment we have only shown that it is possible to combine different multi agent systems. However, as a single simulation is represented by an atomic group with the associated roles and transition functions in the *AGENTMAP* model, it already defines the necessary interfaces of each sub simulation and the available possibilities for interaction.

To form a complete, aggregated simulation, an instance of the related *AGENTMAP* model is now utilized as the master-simulation, thereby defining the structure of the hierarchically interacting, atomic sub simulations. If an aggregated simulation terminates (i.e. that for a finite set of inputs there is a result in finite time [23]), then its result is simply given by the information protocoled during each run of the involved sub simulation.

The termination of the *AGENTMAP* model is out of the scope of this paper. However, it relates closely to the question of termination in agent-based simulations in general.

IV. State of the art

The research domain of agents, multi agent systems and agent-based simulations is relatively young in contrast to other research areas. While in a short time quite a lot of results were published, literature also reflects the inconsistent use of the agent concept and its definition [5]. A survey of actual projects and application areas for agent-based simulations is provided by [24]. It explains also the application of agent-based simulations on business decision problems.

A strictly focused look at agent-based simulations and how to use them was published in the work of Fishwick [25] and Macal and North [26], [27]. The last provides also a tutorial how to build your own agent-based simulations. A comparison with other simulation techniques was carried out by [3] and helps to decide, whether an agent-based simulation is the correct simulation technique to apply.

A summary of methodical approaches for the development of agent-based simulation offers [4] and introduces also a new methodology named *Role-Play-Game*. Another role-based methodical approach was established, as already mentioned, by Ferber and Gutknecht with the *Agent/Group/Role* meta-model [16]–[18]. Odell et al. added a dynamic role assignment [19] and a technical realization [21]. *Normative organization models* [28] have been recommend to restrict the interaction between agents [29]. But the models specifies only interaction protocols and the restriction is not intuitively. Altogether normative organization models are to mathematically and near on implementation to provide an intuitive modeling.

Another interesting model is the fundamental model of *ISLANDER* [30] the so called *Electronic Institution* introduced by [31]. The focus lies on the socieal aspects of the agents and how they communicate together. Their implementations are not relevant. It sounds like the definition of an agent in the *AGR* model. Also there exists roles and scenes, which

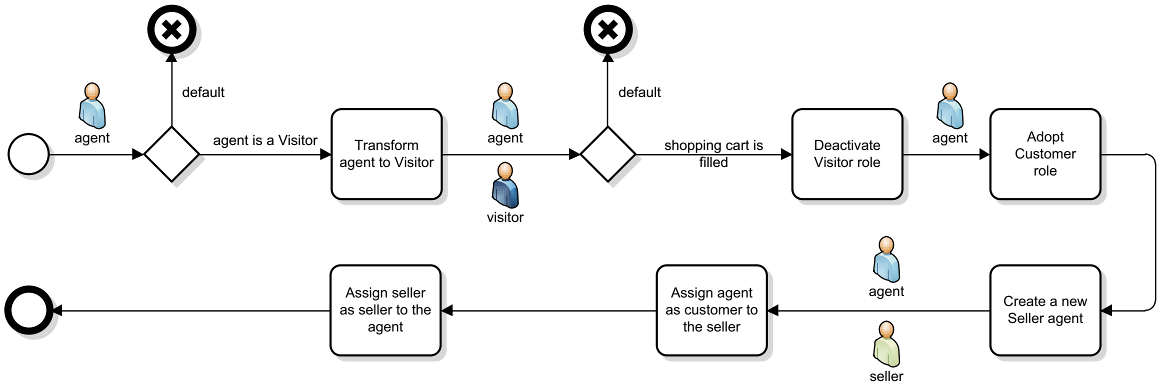

Fig. 10: The transition function of the edge between the product choice and payment process implemented in BPMN 2.0

are not so different from *groups*. Another parallelism to the *AGENTMAP* model is, that they define inter- and intra-scene interaction. But this model is also to mathematically to be understandable for agent technology newcomers.

Complex role models were introduced in literature, together with the necessary algorithms to derive a class model from them [32].

As far as we are aware, group changes have been discussed only in form of migration processes in the context of *mobile agents* [6], [33] so far.

The *AGENTMAP* meta-model presented in this paper is, thus, a logical further step towards a methodical approach in multi agent system development, relating to and building on role-based agent approaches and the mobile agent concept. It also introduces a new application domain for agent-based systems by utilizing them as a backbone technology in the orchestration of distributed simulations.

V. Conclusion

We identified specific problems in combining atomic sub simulations into an aggregated master-simulation, see section I, and used this analysis to structure our approach. By encapsulation all simulations as an agent-based system, we managed to form a consistent view of the problem domain at hand. As a consequence, it was possible to reuse existing models to combine atomic agent-based systems (simulations) into multi agent systems (an aggregated master-simulation) by building on an *Agent/Group/Role* concept. As our main contribution we introduced the *AGENTMAP* meta-model, which restricts and guides the flow of necessary interactions via a so called agent map.

While derived from a life project and real needs, the *AGENTMAP* model is currently still a conceptual framework. It already enables abstract and realistic modeling, but lacks a full blown technical implementation that would yield an executable result.

There is also work left to formally verify the *AGENTMAP* model. However, based on the results in our project *SimProgno*, we believe the model could be a major contribution in the flexible and efficient combination of single simulations into a useful master-simulation. The results, as achieved in the ongoing project, indicate specific strength in managing interfaces, flow integration and intuitive modeling for the end user.

Acknowledgment

This paper was written in the context of the *SimProgno* research project (support code: 01IS10042B) funded by the German Federal Ministry of Education and Research. A part of the illustrations was taken from the technical report [20].

Special thanks to Volkmar Schau, who helped us with his never ending knowledge about agents, Arndt Döhler, who leads the project and gives it the right direction, all other members of the *SimProgno* project and finally to our families especially.

References

[1] A. Hummel, H. Kern, and A. Döhler, "An agent-based simulation of payment behavior in e-commerce," in *Multiagent System Technologies*, ser. Lecture Notes in Computer Science, F. Klügl and S. Ossowski, Eds. Springer Berlin / Heidelberg, 2011, vol. 6973, pp. 41–52, 10.1007/978-3-642-24603-6_6. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24603-6_6

[2] D. Pawlaszczyk, *Skalierbare agentenbasierte Simulation: Werkzeuge und Techniken zur verteilten Ausführung agentenbasierter Modelle*, 1st ed. TU Ilmenau Universitätsbibliothek, Oct. 2009.

[3] P. Davidsson, "Multi agent based simulation: beyond social simulation," in *Proceedings of the second international workshop on Multi-agent based simulation*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001, pp. 97–107. [Online]. Available: http://portal.acm.org/citation.cfm?id=369837.369846

[4] A. Drogoul, D. Vanbergue, and T. Meurisse, "Multi-agent based simulation: Where are the agents?" in *Multi-Agent-Based Simulation II*, ser. Lecture Notes in Computer Science, J. Simão Sichman, F. Bousquet, and P. Davidsson, Eds. Springer Berlin / Heidelberg, 2003, vol. 2581, pp. 43–49, 10.1007/3-540-36483-8_1. [Online]. Available: http://dx.doi.org/10.1007/3-540-36483-8_1

[5] M. J. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. John Wiley & Sons, May 2009.

[6] P. Braun and W. Rossak, *Mobile Agents. Basic concepts, mobility models, & the Tracy Toolkit*, 1st ed. San Francisco, USA: Morgan Kaufmann Publishers/dpunkt.verlag, 2005.

[7] D. Pawlaszczyk, "SIMJADE - Ein Simulationskernel für die verteilte Simulation agentengestützter Modelle," *Tagungsband zum 9. Symposium Soft Computing "Softwareagenten und Soft Computing im Geschäftsprozessmanagement"*, pp. 129–144, Nov. 2006.

[8] G. Tecuci, *Building Intelligent Agents: An Apprenticeship, Multistrategy Learning Theory, Methodology, Tool and Case Studies: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*, 1st ed. Academic Press, Jun. 1998.

[9] P. Knirsch and I. J. Timm, "Adaptive multiagent systems applied on temporal logistics networks," in *In: Logistics in the Information Age. Proceedings of the 4th International Symposium on Logistics (ISL-99) (Muffatto, M., Pawar, K.S. (Eds))*, 1999, pp. 213–218.

[10] N. R. Jennings and M. J. Wooldridge, *Agent technology: Foundations, Applications, and Markets*. Berlin, Germany, 1. Neuauflage nach Original von 1998: Springer Verlag, Feb. 2010.

[11] S. J. Russell and P. Norvig, *Artificial intelligence: A Modern Approach*, 3rd ed. Prentice Hall, Feb. 2010.

[12] M. P. Singh, "A social semantics for agent communication languages," in *Issues in Agent Communication*. London, UK: Springer-Verlag, 2000, pp. 31–45. [Online]. Available: http://portal.acm.org/citation.cfm?id=646973.713683

[13] Y. Labrou and T. Finin, "Semantics and conversations for an agent communication language," in *Readings in agents*, M. N. Huhns and M. P. Singh, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 235–242. [Online]. Available: http://portal.acm.org/citation.cfm?id=284860.284907

[14] P. Bretier and D. Sadek, "A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction," in *Intelligent Agents III Agent Theories, Architectures, and Languages*, ser. Lecture Notes in Computer Science, J. Müller, M. Wooldridge, and N. Jennings, Eds. Springer Berlin / Heidelberg, 1997, vol. 1193, pp. 189–203, 10.1007/BFb0013586. [Online]. Available: http://dx.doi.org/10.1007/BFb0013586

[15] J. Choi, S.-K. Park, S.-C. Baeg, M.-W. Jang, G.-L. Lee, and Y.-H. Lim, "Message-based agent communications in a tightly coupled multiagent system," in *Fourth Golden West International Conference on Intelligent Systems (GWICS-95*, 1995, pp. 194–198.

[16] J. Ferber and O. Gutknecht, "A meta-model for the analysis and design of organizations in multi-agent systems," in *Proceedings of the 3rd International Conference on Multi Agent Systems*, ser. ICMAS '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 128–. [Online]. Available: http://portal.acm.org/citation.cfm?id=551984.852257

[17] ——, "Operational semantics of multi-agent organizations," in *6th International Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL),*. London, UK: Springer-Verlag, 2000, pp. 205–217. [Online]. Available: http://portal.acm.org/citation.cfm?id=648206.749596

[18] O. Gutknecht, J. Ferber, and F. Michel, "Integrating tools and infrastructures for generic multi-agent systems," in *Proceedings of the fifth international conference on Autonomous agents*, ser. AGENTS '01. New York, NY, USA: ACM, 2001, pp. 441–448. [Online]. Available: http://doi.acm.org/10.1145/375735.376410

[19] J. Odell, H. V. D. Parunak, S. Brueckner, and J. Sauter, "Changing roles: Dynamic role assignment," *Journal of Object Technology*, vol. 2, no. 5, pp. 77–86, Sep. 2003, (column). [Online]. Available: http://www.jot.fm/contents/issue_2003_09/column7.html

[20] T. M. Prinz and W. R. Rossak, "Konzeptionelles Meta-Modell miteinander interagierender Simulationen mit rollenbasierten Multiagentensystemen," in *Jenaer Schriften zur Mathematik und Informatik*. Jena, Thuringia, Germany: Friedrich Schiller University Jena, 2012, no. 202012, no. 1/12, Technical Report.

[21] J. Odell, M. Nodine, and R. Levy, "A metamodel for agents, roles, and groups," in *Agent-Oriented Software Engineering V*, ser. Lecture Notes in Computer Science, J. Odell, P. Giorgini, and J. Müller, Eds. Springer Berlin / Heidelberg, 2005, vol. 3382, pp. 78–92, 10.1007/978-3-540-30578-1_6. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30578-1_6

[22] T. Juan, A. Pearce, and L. Sterling, "Roadmap: extending the gaia methodology for complex open systems," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, ser. AAMAS '02. New York, NY, USA: ACM, 2002, pp. 3–10. [Online]. Available: http://doi.acm.org/10.1145/544741.544744

[23] M. Broy, *Eine grundlegende Einführung: Band 1: Programmierung und Rechnerstrukturen*, 2nd ed. Springer Verlag Berlin Heidelberg, Nov. 1997.

[24] A. Deckert and R. Klein, "Agentenbasierte Simulation zur Analyse und Lösung betriebswirtschaftlicher Entscheidungsprobleme," *Journal für Betriebswirtschaft*, vol. 60, pp. 89–125, 2010, 10.1007/s11301-010-0058-6. [Online]. Available: http://dx.doi.org/10.1007/s11301-010-0058-6

[25] P. A. Fishwick, "An introduction to opensimulator and virtual environment agent-based m&s applications," in *Winter Simulation Conference*, ser. WSC '09. Winter Simulation Conference, 2009, pp. 177–183. [Online]. Available: http://portal.acm.org/citation.cfm?id=1995456.1995486

[26] C. M. Macal and M. J. North, "Agent-based modeling and simulation: Abms examples," in *Proceedings of the 40th Conference on Winter Simulation*, ser. WSC '08. Winter Simulation Conference, 2008, pp. 101–112. [Online]. Available: http://portal.acm.org/citation.cfm?id=1516744.1516770

[27] ——, "Tutorial on agent-based modeling and simulation part 2: how to model with agents," in *Proceedings of the 38th conference on Winter simulation*, ser. WSC '06. Winter Simulation Conference, 2006, pp. 73–83. [Online]. Available: http://portal.acm.org/citation.cfm?id=1218112.1218130

[28] J. Hübner, O. Boissier, R. Kitio, and A. Ricci, "Instrumenting multi-agent organisations with organisational artifacts and agents," *Autonomous Agents and Multi-Agent Systems*, vol. 20, pp. 369–400, 2010, 10.1007/s10458-009-9084-y. [Online]. Available: http://dx.doi.org/10.1007/s10458-009-9084-y

[29] O. Boissier, F. Balbo, and F. Badeig, "Controlling multi-party interaction within normative multi-agent organizations," in *Proceedings of the 6th international conference on Coordination, organizations, institutions, and norms in agent systems*, ser. COIN@AAMAS'10. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 357–376. [Online]. Available: http://dl.acm.org/citation.cfm?id=2018118.2018141

[30] M. Esteva, D. de la Cruz, and C. Sierra, "Islander: an electronic institutions editor," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, ser. AAMAS '02. New York, NY, USA: ACM, 2002, pp. 1045–1052. [Online]. Available: http://doi.acm.org/10.1145/545056.545069

[31] J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra, "Engineering open environments with electronic institutions," *Eng. Appl. Artif. Intell.*, vol. 18, no. 2, pp. 191–204, Mar. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2004.11.019

[32] E. Steegmans, D. Weyns, T. Holvoet, and Y. Berbers, "A design process for adaptive behavior of situated agents," in *Agent-Oriented Software Engineering V*, ser. Lecture Notes in Computer Science, J. Odell, P. Giorgini, and J. Müller, Eds. Springer Berlin / Heidelberg, 2005, vol. 3382, pp. 109–125, 10.1007/978-3-540-30578-1_8. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30578-1_8

[33] P. Braun, I. Müller, S. Geisenhainer, V. Schau, and W. Rossak, "A service-oriented software architecture for mobile agent toolkits," *Engineering of Computer-Based Systems, IEEE International Conference on the*, vol. 0, p. 550, 2004.