

Measuring Complexity of Multi-Agent Simulations – An Attempt Using Metrics

Franziska Klügl

Department for Artificial Intelligence,
University of Würzburg
Am Hubland, 97074 Würzburg

kluegl@informatik.uni-wuerzburg.de

Abstract The variety of existing agent-based simulations is overwhelming. However – especially when comparing agent-based simulation to other simulation paradigms, a reference frame is missing that allows characterizing shortly and discriminating between simulation models. In this contribution, I address this problem by introducing metrics for measuring properties of agent-based simulations for finally being able to characterize the complexities involved in developing such a model.

1 Introduction

Multi-agent simulation forms an innovative modeling and simulation paradigm that possesses a great potential for developing models to on a level of detail and in application areas, where it was not possible neither to formulate nor to handle models before. Due to the intuitive structure of a model based on the analogy between agents and the active elements in the original system, modeling and simulation may become a research and analysis method for domain experts that ignored such approaches before.

However, there are several drawbacks that hinder people from constructing and experimenting with valid and useful multi-agent models [Klügl et al., 2004]. Most of these drawbacks are consequences of the flexible design and possibly high level of detail which is resulting in formal and conceptual arbitrariness. Meanwhile, there is an uncountable set of existing agent-based models which can not be compared directly. Even re-implementation attempts often fail because the documentation of the models is incomplete, although the particular dynamics are depending on modeling decisions on a detailed level.

The actual complexity of a model is hidden which hinders systematic utilizing agent-based simulations. In this paper, we will tackle one approach for characterizing complexity of an agent-based simulation residing in the environmental characteristics, the agent structure and dynamics and the overall organization and interaction design. Instead of discussing the sources for complexity on a coarse level, we will define metrics that can be used for measuring different aspects of agent-based models. The aim of our attempt for defining metrics is

basically to analyze and compare aspects of complexity of agent-based simulation. At this point of research, we are not addressing issues like classical software metrics, like deriving production cost etc.

In the remainder of the paper we will characterize agent-based simulation in general, followed by a short introduction to software and especially the state of art in agent software metrics. Before section 5 gives a list of suggestions for metrics of agent-based simulation models, we shortly introduce examples that we will use for their illustration. After a list of example computations, a short conclusion is given.

2 Agent-based Simulation

Agent-based simulation applies the concept of multi-agent systems to the basic structure of simulation models. Active components are identified as agents and programmed using agent-related concepts and technologies. An agent-based model consists of simulated agents that “live” in a simulated environment in virtual time. The environment may play an important role as it frames the agent behaviors and interactions. The individual environment of an agent may consist of other agents, but also may be enriched with resources or objects without agent characteristics related to flexible, autonomous behavior [Klügl, 2001].

Agent-based simulation forms a very attractive paradigm for several simulation application domains. The most obvious is the simulation of emergent phenomena in social science, traffic, biology, etc. Emergent phenomena are “unforeseen” patterns or global behaviors [Holland, 2000] which are not derivable from properties of its constituents.

Agent-based simulations allow the observation of model dynamics on at least two levels: the agent level and the global level. Locality of interaction can be based on explicit representation of space or on abstract non-spatial relationships.

The potential of agent-based simulation is not only restricted to emergent phenomena or self-organization studies. It also forms a elegant basic paradigm for variable structure models [Uhrmacher, 1996]. Additionally, systems that are quite successfully treated using traditional methods can be modeled using agent-based simulation in a more precise and detailed way. A good example is the influence of human workers onto production throughput, especially integrating “intelligent” strategies to cope with non-standard situations.

Due to its attractiveness, the number of available and published agent-based simulation became uncountable. Thus, a kind of language is needed for characterizing the models and their respective complexity in a concise way.

3 Metrics for Agent-based Software

3.1 Metrics in General Software Engineering

The idea of measuring general properties of software has been attracting researcher for many years. Software metrics hereby can be seen as the mapping

from a piece of software to the domain of numbers. Such functions characterize certain properties concerning size, complexity, cost, design, etc.

The mostly known metric in software engineering, and basically the one that is generally used, is the Lines Of Code (LOC) metric that forms the basis for different heuristics about duration of implementation/modifications, cost, overall error probabilities, etc. A description of such conventional metrics can be found in [Conde et al., 1986] or [Thaller, 2000]. They also give an introduction to the Halstead metrics that are based on operator and operand numbers for predicting program volume and effort. However, their expressiveness is discussed controversially.

Also in the simulation area, early attempts for defining metrics for simulation models have been tried. J.C. Wallace [Wallace, 1987] e.g. suggests a “Control and Transformation Metric” that basically consists of counting input/output variables per node combined with the number of nodes in a graphical representation.

As multi-agent systems are often implemented using/based on object-oriented programming languages, metrics for those languages may be useful also in the agent context. However, as expected, structural metrics like weighted methods per class, depth of inheritance tree or coupling between object classes etc., like suggested in [Chidamber and Kemerer, 1994]. These and metrics that focus on the coupling between classes ([Briand et al., 1997]) seem to be too low-level for being meaningful for agent-based software, as well as for agent-based simulation models.

3.2 Metrics in Agent-based Software Engineering

From sources of complexity in agent-based system design, like discussed in [Wooldridge, 2002], only a small step seems to be necessary to suggestions of measuring complexity. Indeed, there are several suggestions. The work of Wille and Dumke and co-workers ([Wille et al., 2004], [Dumke et al., 2000]) seems to be the most extensive; however they only give a long list of informal metric suggestions without detailing the way for computing them. [Far and Wanyama, 2003] introduce metrics for measuring agent complexity in order to facilitate system decomposition based on a survey of sources of complexity for agent-based systems. Gómez-Sanz et al. [Gómez-Sanz et al., 2006] focus on cost estimation. They identify different descriptive variables, e.g. number of rules or number of state machines for characterizing behavior or number of mental entities or number of goals for describing the informational complexity of the agents. They also relate these variables to the LOC metric based on data from three EU projects. The systems are also treated using metrics for early development phases. Metrics are actually applied resulting in figures that are also related to actual costs.

Particular metrics for measuring performance of organizational design were suggested in [Robby et al., 2006]. Another example for the use of specific metrics in agent-based system engineering is [Woodside, 2001] that evaluates scalability of systems of mobile agents.

However, agent-based simulation models can not be treated like most other agent software at least for two reasons: the first is the relevance of the simulated environment [Klügl et al., 2005], the second is determined by the relation with a reference or original system that has to be guaranteed. Thus, simplicity of a model is essential as a minimal set of assumptions is a prerequisite for feasible validation. The ability for comparing model complexity is thus central for evaluating model design.

4 Example Application

Before presenting our suggestions for metrics, we shortly give information about models that we will use as examples for illustration throughout the metric introduction and later to test the metrics.

Our first example is the Sugarscape model [Epstein and Axtell, 1996]. Agents move over the discrete Sugarscape map, locally searching and harvesting sugar. Sugar is a renewable resource carried by every cell characterized by a maximum capacity. Only the current sugar stock is perceivable by the agents within some individual range. The cells are arranged so that two sugar hills exist. Agents possess a sugar-consuming metabolism and personal sugar storage. They always move to the next perceivable cell with maximum sugar stock that is not yet occupied. Agents may starve or die of some age limit and produce offsprings. This is the model that is described in the first chapters of the Epstein & Axtell book. In later chapters, this model is enhanced by a second resource, named spice, for modeling trade and other social phenomena. We will only deal with the basic versions of the model. To show how small details affect the metrics we compare two slightly different stages of the Sugarscape model: a first one without maximum age and reproduction, the second with maximum age and sexual reproduction.

The second example we are using for illustration, is the Tribute model of R. Axelrod [Axelrod, 1995]. Activated agents evaluate the vulnerability of direct neighbors or neighbors of committed agents for deciding whether to demand tribute or not. The addressed agent evaluates the costs of fighting and decides for paying tribute or fighting. These actions have effect on the wealth of the agent and on their commitment to each other, later influencing their decision making. A central environment distributes a small value of wealth as a basic income after three activations.

These two models can be seen as basic prototypes for agent-based simulations. The Sugarscape model resembles a very simple land-use-type model where agents interact with their local environment in the first instance and only secondly with other agents depending on the resources they have acquired from their environment. Interaction is mainly mediated by the environment. The Tribute model can be seen as a representative for a second kind of models that focus on interaction-induced structures. Locality is modeled based on agent-agent relations resulting in network-type structures. Interactions between agents produce

some legacy in the mental structures of the agents that again influences future interactions.

The third example we use for demonstrating is the SBBpedes model. It was developed for a large simulation study of the pedestrian behavior in the SBB railway station Bern [Klühl and Rindsfuser, 2007]. It is particular as the agent behavior has integrated also simple planning activities beyond pure locomotion as in most other pedestrian simulations. In contrast to the two previously sketched models, it is a model used in a successful real-world application. Agents determine their final destination as far as possible (e.g. they cannot select the door of the train they will board when the train did not yet arrive at the platform) and then construct a coarse plan on the area-level (which stairway to take ...). This plan is executed using some standard collision-free locomotion model. Depending on the current situation, a simulated pedestrian adapts its individual coarse plan or uses intermediate destinations on a lower level for bypassing obstacles, etc. The simulation reproduced the real situation during the most busiest morning hours with about 80 trains on 12 platforms and all together about 45000 boarding, alighting and transferring travelers. It was used for evaluating layout alternatives and a planned change in train schedule and platform assignment.

5 Suggestions for Metrics

Even this small list of three possible agent-based simulation models shows that there are a variety of dimensions for characterizing them – starting from environmental structures, to organizational forms, agent architectures, but also in more technical terms of agent interaction, communication language... Thus, the question arises, whether considerations about detailed characterizing, but informal descriptions may help for characterizing the complexity involved in the design and implementation of such an agent-based simulation. Metrics have the advantage of being objective and exact – if they are appropriate for capturing the intended properties. Model metrics – in analogy to software metrics – can be seen as functions that map the model to a numeric value that characterizes some property of the model.

What does complexity of an agent-based model in relation to a user (modeler, stakeholder, domain expert, etc.) mean? Basically it consists of understandability for the human and is connected with the predictability of the model dynamics and output. Understandability means clarity of structures and relations. It is also influenced by size and heterogeneity of the individual agents as well as of the overall system. Predictability refers to the effort and skills of the modeler needed for traceability of behavior and interactions.

5.1 Automatically Computable Metrics

Based on these thoughts, we can identify the following metrics. We may distinguish between overall system-level metrics (including metrics for measuring the

complexity of the environment) that are relevant for the complete model, agent-level metrics and agent-system-level. The latter involve interactions or relations between agents.

On the overall system level one may tackle metrics that are not surprising, starting from the population sizes and their dynamics. However, even those metrics are not trivial as their values are scenario-dependent. That means, the metrics can only be used to characterize one particular, completely specified simulation run, for characterizing a complete model, especially with stochastic elements - means over more than one run have to be used.

NAT: Number of Agent Types is a measure for heterogeneity of the model.

It basically resembles the number of classes like an object-oriented metric and can be easily computed in simulation models, respectively in their implementations. The highest heterogeneity occurs when all agents are structurally different, i.e. NAT equals the number of agents. However, typification is not the only way to represent heterogeneity among agents. Thus, this measure may not be sufficiently valid to characterize all forms of heterogeneity. For example, if only parameters like thresholds or weights are set individually, but all agents structurally belong to the same class, then $NAT = 1$, although the effectively observable behavior of all agents is different.

NRT: Number of Resource Types like the NAT-Metric, but for passive entities of the environment.

MNA: Maximum Number of Agents is probably the most obvious measure for the size of the model – the maximum number of agents concurrently present during a simulation run. However, there is a conceptual problem when the maximum number is only adopted at the beginning of the simulation. This happens e.g. when the question is tackled how many agent can be subsisted by a particular environment. This is already happening in simulations like Sugarscape. In such cases, one may doubt the meaningfulness of such a measure as the number of agents is intentionally set too high. The number of agents to which the simulation is converging to, would make more sense. Another idea may be to use the sum of agents that are existing in the simulated environment over the complete simulation time. In the SBBpedes example, the maximum number of agents concurrently tackled is about 9000, the overall number sums up to about 45000. Also this difference accents the dynamics of a system.

MNR: Minimum Number of Resources is the analogue for the maximum number of agents. Here, one may argue that in particular, only resources should be counted that may be actually used by an agent – see e.g. in the Sugarscape world there are 2500 cells, but a not negligible share of them does not carry any sugar (cell capacity equals zero). In some simulations, inanimate objects are used for decorating the environment in order to produce nice animations. Whereas in the first example the relation between cells with and without sugar may be interesting, decoration elements should be ignored.

MDA: Maximum Delta of Agent Population is a measurement for determining the variability of population numbers over a given interval of time, typically one simulation step. Thus, it forms the rate of population change. In models that contain probabilistic aspects related to agent lives, the actual dynamics may vary from time interval to interval as well as between runs. Also here, the initial phase with potentially higher death rates should be distinguished from the converged state.

MDR: Maximum Delta of Resource Population is the analogue to the MDA metric.

ARR: Agent-Resource Relation is the number of agents divided by the number of resources. Here mean and variation is interesting.

NASH: Number of Agent Shapes This is a measure for spatial complexity. How many different geometries may agents possess? This makes only sense in simulation with map-based spatial representation.

NRSh: Number of Resource Shapes is the analogue to the NASH metric: How many different geometries do occur in the set of resources?

MRS: Maximum Resource Status Size Resources may be differently complex. Whereas obstacles may only possess purely spatial attributes like extent, form and position. Others may carry more sophisticated information. This metric counts the maximum number of status variables a resource may possess. The question what are status variables, may arise. In the simple Sugarscape model, a cell may possess one status variables. However, every cell needs two additional parameter, namely the growth rate and maximum sugar capacity. Although the latter two influence the status, they are parameter, no state variables. The state of a resource may also be used as data container for agents; for example in the SBBpedes simulation (see below) a train is a resource that carries data about length or number of doors which would be counted as parameters. But, a train is also used to collect information about its travelers. How many have currently arrived at their goal? Information like this forms the status of a train resource.

MRP: Maximum Resource Parameter This metric computes the maximum number of parameter that influence the values of the status variables. Following the examples of MRS, the growth rate or maximum stock form parameters. Other examples are initial values. In other scenarios, resources carry (static) information used not to update the status of the resource, but used by agents to guide their behavior according to this information. This constant information items are also subsumed under this metric.

Whereas the metrics above aim at measuring population and environmental complexity, the agents and their interactions naturally form a source of complexity for an agent-based model that is worth of being measured. However, evaluating agent complexity with quantitative values results in more sophisticated metrics than those above. All following metrics are measures for individual agents. Thus, for characterizing a complete model, they have either to be aggregated or computed for a “typical” agent. Aggregation can consist of averaging using the maximum or minimum over the complete agent population, or simply summing up.

ACR: Architectural Complexity Rank Complexity of the agent architecture might be a reasonable measure. Unfortunately, indicators for it are not obvious. We suggest to simply rank the architectures into three sets along their complexity and use this as a metric.:

1. **Behavior-describing architectures** are all rule-based structures that aim at reproducing individual behavior based on directly describing it. They do not claim to resemble actual cognitive processes of decision making but are more like a black box description of observed behavior. Examples are rule- and activity-based descriptions of behavior.
2. **Behavior-configuring architectures** are quite common in agent-based simulation as they form a flexible goal- or utility-based architecture with efficient reasoning based on task- or activity representations like skeletal plans. Actually, this is the category of BDI architectures.
3. **Behavior-generating architectures** are using traditional AI planning. The behavior of an agent is actually planned by an agent based on operator representations with pre-conditions and post-conditions. The agent generates a sequence of actions leading to its explicitly represented goal.

APM: Action Plasticity Metric For being really sensitive, the ACR metric has been combined with additional measures concerning the behavioral plasticity and variability. Plasticity denotes the potential adaptivity of behavior in reaction to environmental influences. That means predominantly the extent of the behavioral repertoire and the flexibility in its application: For discrete actions, this metric is computed by simply counting possible actions. When actions are parameterized, the range of the parameters has to be multiplied. As an illustration take the following example of a simple pedestrian simulation: the agents may move with a standard speed. In addition, they have the possibility to turn in reaction to obstacles. The angle is a parameter for the turn action. If e.g. only turning actions with an angle of 45° and 90° in both directions are allowed, this action space metric would return $1 + 1 * 4 = 5$. If the angle has a continuous range, the metric would return ∞ . Unfortunately, an additional continuous parameter would not affect the outcome of the metric. In this case, it could be more descriptive to introduce an additional metric describing the basic types of actions. Yet, the idea of APM consists in denoting the most basic degree of freedom in action selection.

SPK: Size of Procedural Knowledge Another metric related to behavior plasticity is the size of the procedural knowledge that is available for an agent. Its computation must be dependent on the particular form of architecture.

The first idea would be counting possible extensions of plan skeletons in behavior-configuring architectures. However, behavior-describing as well as the behavior-generating architectures are difficult to characterize in a corresponding way: the number of possible rule or operator chains would be a comparable measure. But how to compute it? Its determination is depending on the life span of agents, whether the environment poses episodic behavior,

etc. In worst case, one has to suppose that at every point of time, every rule in the behavior description of an agent can be applied.

Thus, we reduce the computation of SPK to the following computations: In behavior-describing architectures, it equals the number of rules that define the agent behavior, in behavior-configuring architectures, the number of plan skeletons is counted, including explicitly represented partial plan skeletons. In these two cases, a set of additional metrics is useful for characterizing the complexity of the rules or plan skeletons, as the rules, as well as the plan skeletons may be differently complex. These metrics may count the number of conditions, generality of conditions, number of branching elements in the skeleton, etc. Metrics for rule-based systems were developed in the early 90ies, see for example [Chen and Suen, 1994].

The computation of the SPK of behavior-generating architectures also needs some discussion: The number of possible action sequences is the first idea for a definition, but would be not comparable to the number of plan skeletons as the latter may contain more than one paths per skeleton according to conditioned expansion in hierarchical representations. Also, the number of operators would not be a good measure, as it does not represent the potential complexity of the procedural knowledge of the agent. Despite of the potential combinatoric explosion, there seems to be no other reasonable way than to define the SPK for behavior-generating architectures as the number of possible action sequences.

NCR: Number of Cognitive Rules The share of actions that affect the internal beliefs or status of an agent. One may also denote these as cognitive rules. Updates of mental models can be an interesting indicator for the reasoning complexity of the agent, although it is ignoring the variety of used algorithms. At least, one may derive a measure for independence of actions from that information – as far as the environment is not used as an external memory. However, in general the usefulness of this metric can be doubted. Often, mechanisms for updating the mental model use different procedural structures than the actual action selection procedure. Thus, we leave the problem of formulating a metric general at this point for future research.

Interactions between agents are highly dynamic. It is questionable if they can be handled in an expressive way at all beyond pure message counting. Whereas the measures above can be computed based on static model code, the values of the following measures can be determined during a simulation run.

SPII: Sum of Public Information Items A good measure for the size of external interfaces seems to be the number of concurrently publicly accessible variables or information items. In the Tribute model, the wealth and commitments of every agent is common knowledge. That means, every agent knows about the wealth and all the commitment status of every other agent. Consequently, we have a value of $1 \times n + n \times (n - 1)$ as the available information items for a single agent; As the status is equal for every agent,

the resulting metric returns 100 for $n = 10$. Sugarscape [Epstein and Axtell, 1996] is another example that illustrates the dilemma of this approach. Here, interaction is strictly local. Every agent interacts only with its local neighbors or the cells within its perception radius. It can only perceive the current sugar stock of such a cell. Thus, there is no global knowledge, but information is in principle publicly accessible – just restricted to those who are there. In the very first scenarios, agent may just perceive the sugar storage of cells within their range – which consists of an area of $k \times k$ cells. With $k = 6$ and 300 agents - this would mean that for the complete agent population 10800 data items are concurrently available.

This kind of metric becomes more meaningful, if we divide this value by the number of shared data units. In the Tribute model this would result in the same value as all data units are accessible for all agents at every point in time. The sugarscape model we have to divide it by the number of all available status units. This results in $10800/2500 = 4.32$ - basically this means that with the initial agent numbers the intersection between two sets of perceived sugar cells contains in the mean a little bit more than 4 cells. However, after only a few steps, the population is decreasing and concentrating on the cells with higher sugar values. In a population of only 50 agents, this measure would result in a value of $50 \times 6/2500 = 0.12$. When the relation between environmental information and agent needs is lower than 1, it indicates that the perception radius of agent do not intersect in mean. However, as the agents concentrate on a small region, this measure might be misleading.

One might think, that this metric does not work for purely message-based multi-agent systems. However, it is a question of abstraction. The SPII metric deals with information units independently whether they are transferred via the environment or message-based.

NEA: Number of External Accesses In addition to the number of available information units, an interesting property is how often external data is accessed by the agent in its behavior definition. Basically this is an abstraction from some message counting metric. Especially together with the SPII metric, these metric promises to form an interesting measure for the amount of external information that the agent may actually processes per time step. It nicely discriminates between highly interactive simulations and models where the agents only once access information and then process these potentially outdated information.

NAR: Number of Agent References A metric addressing the coherence of the agent system is the mean number of agents contained in one agents internal model: This is basically a measure for the degree of connection within the agent system. As this value may be varying over time, we may distinguish between NAR-mean and NAR-stdev. Also, minimum and maximum number of references as well as the time-related delta of these values may be interesting as it indicates the dynamics of the system in terms of relations between agents.

NRR: Number Resource References The number of references of an agent holds towards addressing resources. Using this metric, we can distinguish between models contain more or less detailed elements of ownership.

NMA: Number of Mobility Actions This metric only makes sense when there is an actual map where the agents may change their local position and thus surroundings. It is measured in number of move actions per agent per time step. In combination of the SPII and NEA metrics, it shows the dynamics of relations.

This compilation of suggestions for metrics in agent-based simulations covers a variety of relevant aspects, yet is far from being complete. Metrics quantifying aspects of protocols and conversations are missing. Number of conversations, mean number of message per conversation, etc. Such metrics would support the higher level description of interactions. Another area that is under-represented is a distinction according to different relations (acquaintance, dominance,...) between the agents – number of relational categories ...

One aspect that complicates the computation of relevant quantities are variations of a model for experimentation. Often the number of agents and resources, etc. are modified during experimentations, the maximum number of agents depends on the concrete environmental conditions of the scenario, etc.

Most of the metrics given tackle pure size of a model. The complexity of designing, treating and understanding of a model lies in the existence and transparency of feedback loops. However, it is even hard for a human being to identify existing feedback loops in a piece of simulation specification or especially in a already coded model.

5.2 Tackling Feedback Loops?

Although the metrics above are attractive due to the possibility of automatic, non-human-done computation, one notices that they are able to capture mostly subordinate features - the core properties that determine complexity, understandability, etc. of a multi-agent model are not addressed by them: How to count feedback loops?

Even for humans, the existence of feedback loops – especially multi-level loops – is hard to determine just based on a static model specification or implementation. Hidden feedback loops form the backbone for every complex problem. Every change of a status value, every interaction can be part of a feedback loop.

Thus, the number of positive and negative feedback loops, sub-divided into one-level and multi-level feedback loops can be determined by a human knowing about the contents and design of the model. However, as mentioned before the identification of a feedback loop is a complex task, especially without the necessary variety of instruments for capturing the core dynamics in an abstract way beyond message counting.

5.3 Language-Specific Metrics

When using traditional programming languages for implementing an agent-based simulation only general metrics can be applied. If the simulation is based on particular tools and architectures, more specific and meaningful metrics can be defined. This specially applies to the APM (Action Plasticity Metric), the SPK (Size of Procedural Knowledge) and the NCR (Number of Cognitive Rules) metrics that tackle details about the decision making and behavioral flexibility of the agent. Without reference to specific architectures, the identification of such variables is quite hard. Sometimes, it even violates the requirement of objectivity and automatic computation.

Thus, in an agent-based system implemented using the JADE framework (jade.tilab.com), the number of “behaviors” of agent may be interesting. Such “behaviors” form here the basic structure for behavior definition. Also for agents using the PRS architecture [Ingrand et al., 1992] or one of its legacies like JACK (www.agent-software.com), the number and size of Knowledge Areas per agent determines the complexity and sophisticated-ness of agent behavior. Similar metrics may be meaningful for agents designed based on the RAP architecture [Firby, 1989]. Analogous metrics may be found in any agent system and simulation that is based on some form of high-level structure.

This is also the case with SeSAm (www.simsesam.de) which is used as implementation basis for all example computations in the next section. We did this to avoid tampering based on different implementation styles. In SeSAm, the behavior of agents is structured along a graph, named “reasoning engine” that contains activities – which are some form of script – and rules that are used for controlling the transition between activities. The state of an agent consists of a set of state variables with potentially complex data structures. Thus, in SeSAm, among others, the number of parallel reasoning engines, number and size of activities and rules per graph, number of variables, may provide interesting measures of the size and complexity of a model. A set of specific metrics for SeSAm has been suggested in [Bülow, 2005].

5.4 Test and Assessment

One may notice that the use of metrics in standard software engineering is actually hardly established and their usefulness is still subject to discussion.

For demonstrating potential of metrics for agent-based simulation, we want to give some example computations for the models shortly described in section 4. For reducing effects of potentially hidden implementation details, all models were implemented using the same simulation environment: We used the above mentioned SeSAm, as it provides a convenient high-level languages combined with visual programming. The main reason was, that we were quite familiar with the modeling facilities provided by it. Thus, re-implementation implied minor effort for the Sugarscape and Tribute model. The SBBpedes project was originally done using SeSAm.

| Metric | Sugarscape I | Sugarscape II | Tribute | SBBpedes |
|---------------|-------------------|---------------|---------|-----------------------|
| NAT | 1 | 1 | 1 | 5 |
| NRT | 1 | 1 | 0 | 5 |
| Initial NA | 300 | 300 | 10 | 110 |
| MNA | ca. 15 (conv.) | ca 1350 | 10 | ca. 9000 |
| MNR | 2500 | 2500 | 0 | 140 |
| MDA | -13, +0 | -40, +74 | -0, +0 | -14,+21 |
| MDR | 0 | 0 | 0 | 0 |
| ARR | 0.12 | 0.56 | indef. | 36 |
| NASh | 1 | 2 | 1 | 1 + 250 |
| NRSh | 1 | 1 | 0 | 250 |
| MRS | 1 | 1 | 0 | 11 |
| MRP | 2 | 2 | 0 | 16 |
| ACR | 1 | 1 | 1 | 1+3 |
| APM | 26 | 27 | 12 | ∞ |
| SPK | 2 | 4 | 4 | |
| NCR | 0 | 0 | 2 | 3 (plan) + 2 (move) |
| SPII | 4.32 | 4.32 | 100 | 0 |
| NEA | 36 (for $k = 6$) | 36 | 9+9=18 | min. 1 |
| NAR | 0 | 2+23 | 9 | 0 |
| NRR | 1 | 1 | 0 | 1 to 8 (planned path) |
| NMA | 1 | 1 | 0 | 1 |
| SeSAm-NA | 2 | 5 | 8 | 23 |
| SeSAm-NR | 3 | 7 | 11 | 59 |

Table 1. Application of metrics onto three example model implementations. All models contain stochastic elements, therefore at some places only rough numbers are given, when the exact number slightly varies between two runs.

The results of our computations are shown in table 1. Hereby, one has to keep in mind that we did not aim at studying the outcome of the model, but we were searching for general measures of complexity for these models.

One may see that the Sugarscape models are quite simple but show an interesting population dynamic. The number of agents is dynamic - with a higher dynamic in variant II than variant I, even higher than in the SBBpedes model. The Tribute model does not possess any form of explicit population dynamics. However, some agents become incapable of acting due to their low wealth. This is not expressed by the current set of metrics - a metric denoting the effective number of active agents would be necessary.

From the point of view of population dynamics and environmental representation and agent-environment interaction, the Tribute model is simple - there is no relevant environment, however it exhibits the highest complexity in available information about other agents (SPII) and possible full connectivity as the NAR-value correspond to the number of agents minus 1 (no self-reference).

The SBBpedes model is larger than the others in terms of mere agent numbers as well as in extend of agent behavior. However, the interaction between agents is comparatively simple. No agent possess explicit information about other agents within its belief model. Direct interactions are seldom.

The main question that remains is - of what use are these numbers? Up to now, the metrics can be used for demonstrating “areas” (in terms of subsets of metrics) of higher complexity relative to other models. We have seen that they are able to discriminate between models. For an absolute complexity measure, the set of isolated metrics has to be re-considered, potentially extended and solicited. Then, these basic metrics have to be weighted and combined resulting in a characteristic that can be used for supporting the management of a simulation study, for estimating simulation effort or for evaluating simulation tools.

6 Conclusion

Despite of a lot of scientific effort, software metrics are still controversially discussed in practice. In this paper we suggested a set of metrics and illustrated them by applying them to a set of existing, and partially well-known models. Although we concentrated on mere size-related metrics, their application allowed to expose details of complexity characterizing the individual models. The metrics also allowed to discriminate between two slightly different variants of the Sugarscape model. Consequently, one may state that these set of metrics seems to be a good starting point towards evaluating and comparing agent based simulation models although several aspects were left to future efforts. The next steps clearly involve the development of more dynamics-related metrics and the application to more simulation models for finally reaching the goal of a short and precise characterization of agent-based simulation model complexity.

References

- [Axelrod, 1995] Axelrod, R. (1995). A model of the emergence of new political actors. In Gilbert, N. and Conte, R., editors, *Artificial Societies: The Computer Simulation of Social Life*, page 19ff. UCL Press.
- [Briand et al., 1997] Briand, L., Devanbu, P., and Melo, W. (1997). An investigation into coupling measures for C++. In *Proceedings of the 1997 (19th) International Conference on Software Engineering*, pages 412–421.
- [Bülow, 2005] Bülow, M. (2005). Metriken für Multiagentensimulationen in SeSAM. Master’s thesis, Institute of Computer Science, University of Würzburg.
- [Chen and Suen, 1994] Chen, Z. and Suen, C. Y. (1994). Complexity metrics for rule-based expert systems. In *International Conference on Software Maintenance, 1994*, pages 382–391.
- [Chidamber and Kemerer, 1994] Chidamber, S. R. and Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Trans. Software Engineering*, 20:476–493.
- [Conde et al., 1986] Conde, S. D., Dunsmore, H. E., and Shen, V. Y. (1986). *Software Engineering Metrics and Models*. Benjamin/Cummings.
- [Dumke et al., 2000] Dumke, R. R., Koeppe, R., and Wille, C. (2000). Software agent measurement and self-measuring agent-based systems. Technical Report 11, Fakultät für Informatik, Uni. Magdeburg.
- [Epstein and Axtell, 1996] Epstein, J. M. and Axtell, R. (1996). *Growing Artificial Societies. Social Science from the Bottom Up*. Random House Uk Ltd.
- [Far and Wanyama, 2003] Far, B. H. and Wanyama, T. (2003). Metrics for agent-based software development. In *IEEE CCECE 2003. Canadian Conference on Electrical and Computer Engineering, May 2003*, volume 2, pages 1297–1300.
- [Firby, 1989] Firby, J. (1989). *Adaptive Execution in Complex Dynamic Worlds*. PhD thesis, Yale University.
- [Gómez-Sanz et al., 2006] Gómez-Sanz, J. J., Pavón, J., and Garijo, F. (2006). Estimating cost for agent-oriented software. In Müller, J. and Zambonelli, F., editors, *Agent-oriented software engineering V. 5th International Workshop, AOSE 2005, Utrecht, The Netherlands, July 2005, Revised Selected Papers*, number 3950 in LNCS, pages 218–230.
- [Holland, 2000] Holland, J. H. (2000). *Emergence. From Chaos to Order*. Oxford University Press.
- [Ingrand et al., 1992] Ingrand, F. F., Georgeff, M. P., and Rao, A. S. (1992). An architecture for real-time reasoning and system control. *IEEE Expert*, 7(6):34–44.
- [Klügl, 2001] Klügl, F. (2001). *Multiagentensimulation – Konzepte, Anwendungen, Tools*. Addison Wesley.
- [Klügl et al., 2005] Klügl, F., Fehler, M., and Herrler, R. (2005). About the role of the environment in multi-agent simulations. In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems*, number LNCS in 3374, pages 127–149. Springer.
- [Klügl et al., 2004] Klügl, F., Oechslein, C., Puppe, F., and Dornhaus, A. (2004). Multi-agent modelling in comparison to standard modelling. *Simulation News Europe*, 40:3–9.
- [Klügl and Rindsfüser, 2007] Klügl, F. and Rindsfüser, G. (2007). Large-scale agent-based pedestrian simulation. In Müller, J. P., Petta, P., Klusch, M., and Georgeff, M., editors, *Multi-Agent Technologies V, Proceedings of the MATES 2007*, number 4687 in LNAI. Springer.

- [Robby et al., 2006] Robby, DeLoach, S. A., and Kolesnikov, V. A. (2006). Using design metrics for predicting system flexibility. In *Proceedings of the 2006 International Conference on Fundamental Approaches to Software Engineering (FASE 2006)*.
- [Thaller, 2000] Thaller, G. E. (2000). *Software-Metriken – einsetzen, bewerten, messen*. Verlag Technik, 2nd edition.
- [Uhrmacher, 1996] Uhrmacher, A. M. (1996). Object-oriented and agent-oriented simulation-implications for social science applications. In Doran, J., Gilbert, N., Müller, U., and Troitzsch, K. G., editors, *Social Science Micro Simulation- A Challenge for Computer Science*, Lecture Notes in Economics and Mathematics, pages 432–447. Springer, Berlin.
- [Wallace, 1987] Wallace, J. C. (1987). The control and transformation metric: Toward the measurement of simulation model complexity. In Thesen, A., Grant, H., and Kelton, W. D., editors, *Proceedings of the 1987 Winter Simulation Conference*, pages 597–603.
- [Wille et al., 2004] Wille, C., Brehmer, N., and Dumke, R. R. (2004). Software measurement of agent-based systems - an evaluation study of the agent academy. Technical Report Preprint No. 3, Faculty of Informatics, University of Magdeburg.
- [Woodside, 2001] Woodside, M. (2001). Scalability metrics and analysis of mobile agent systems. *Lecture Notes in Computer Science*, 1887:234ff.
- [Wooldridge, 2002] Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems*. John Wiley.