2017

Features of Agent-based Models

Reiko Heckel
University of Leicester

Alexander Kurz
Chapman University, akurz@chapman.edu

Edmund Chattoe-Brown
University of Leicester

Recommended Citation

R. Heckel, A. Kurz, and E. Chattoe-Brown, "Features of Agent-based Models," Electronic Proceedings in Theoretical Computer Science, vol. 263, pp. 31–37, Dec. 2017.

Features of Agent-based Models

Reiko Heckel
Department of Informatics
University of Leicester, UK
rh122@leicester.ac.uk

Alexander Kurz
Department of Informatics
University of Leicester, UK
ak155@leicester.ac.uk

Edmund Chattoe-Brown
Department of Sociology
University of Leicester, UK
ecb18@leicester.ac.uk

The design of agent-based models (ABMs) is often ad-hoc when it comes to defining their scope. In order for the inclusion of features such as network structure, location, or dynamic change to be justified, their role in a model should be systematically analysed. We propose a mechanism to compare and assess the impact of such features. In particular we are using techniques from software engineering and semantics to support the development and assessment of ABMs, such as graph transformations as semantic representations for agent-based models, feature diagrams to identify ingredients under consideration, and extension relations between graph transformation systems to represent model fragments expressing features.

1 Introduction

The state of the art in Agent-based Modelling (ABM) is to identify phenomena of interest around which small models are developed to support specific explanations. Examples include the emergence of co-operation in competitive environments, analysed by games of Iterated Prisoner's Dilemma (IPD), or the dynamics of opinion formation in networks, using variations on the Voter model.

Often the design of such models is ad-hoc, in particular when it comes to choosing and validating their ingredient features and calibrating their parameters. For IPD we may consider features such as network structure, decision making, learning, and network change. Opinion formation models can address the impact of media and external events, network structure, network change, cognitive dissonance, psychological factors, strength of conviction, etc. A scientific approach requires both

1. a systematic analysis of the choices faced when creating models, including a methodology to compare and assess their impact;
2. a way to calibrate a model's parameters and validate its results against empirical observations.

Without addressing these requirements, models will only be useful in answering isolated questions, making little contribution to a comprehensive understanding of the phenomena themselves, nor their relation with reality.

In this paper we consider an approach to answering the first question, the systematic analysis of the combinations of features that could be addressed in ABMs. We propose techniques from software engineering and semantics of modelling and programming languages to support the design and assessment of such features, in particular

- graph transformation, as semantic representation for agent-based models
- feature diagrams, to identify ingredients under consideration and state their interdependencies
- extensions extension relations between graph transformation systems, to represent model fragments expressing features

2 Background

Agent-Based Modelling [2], hereafter ABM, is a technique for understanding social phenomena, distinct from both statistical approaches (like regression analysis) and those based on narrative accounts (like ethnography). Its distinctiveness arises from the use of a computer program explicitly representing the cognition and (inter)action of "agents" (simulated social actors) to explore the aggregate effects of these. This gives the usual advantages of formal approaches (when compared to the risk of incomplete, faultily reasoned or contradictory narratives) without the implausible simplifying assumptions often required of such approaches. The approach also gives rise to a distinctive methodology in which assumptions about human behaviour (based on qualitative interviews or experiments for example) can be calibrated independently of the match between real aggregate data and its simulated equivalents (validation). This approach is importantly different from the more common "fitting" of statistical models in which match is not achieved as a falsifiable hypothesis (as it is in ABM) but by deliberate adjustment of model parameters without a robust social interpretation. (The slope of a regression line tells us about a pattern in data. It is not clear if it tells us anything about individual behaviour.) ABM are also important because systems of interacting agents are often complex, leading to counter-intuitive outcomes in aggregate even when individual behaviours are understood. This makes casual inference from statistical regularities to individual behaviour and "grossing up" from individual behaviour narratives to aggregate outcomes potentially unreliable.

Despite the compelling logic of this methodology, however, large numbers of non-empirical (not calibrated or validated) ABM are still published [1]. These often select elements of a social phenomenon arbitrarily (for example social influence, social networks, geography and rationality) which makes them potentially non-comparable as well as non-empirical (see for example [4, 6, 7] as typical examples from the huge ABM literature on the Prisoner's Dilemma). Obviously the most effective way to evaluate these non-empirical models would be using data [3]. Unfortunately, for a variety of reasons (at least some of which are scientifically legitimate) this is not always feasible. Under these circumstances, new tools that allow us to systematically explore and evaluate the space of alternative models are a valuable alternative.

3 Example: The SIR Model

An SIR model is an epidemiological model for the spread of a disease in a population of agents predicting the numbers infected with a contagious illness over time. Fig. 1 shows the most basic version of such a model. Individual agents change state from *S*usceptible to *I*nfected before becoming *R*esistant. Modelled as graph transformation system, agents and their data are specified by a type graph and their actions by graph transformation rules, as shown in Fig. 1. The type graph defines a single node type *Agent* with an attribute s for state that can assume values S, I or R. Rule *infect* describes how an agent can change state from S to I in the presence of another *I*nfected agent. Rule *recover* states that an infected agent can become resilient.

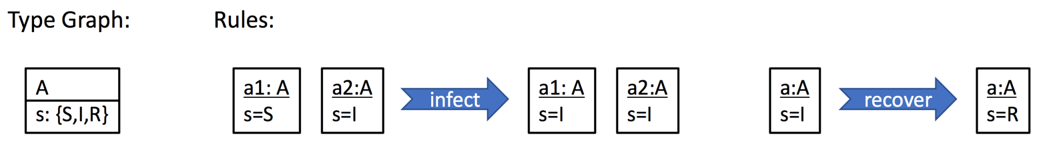

Figure 1: Basic SIR model

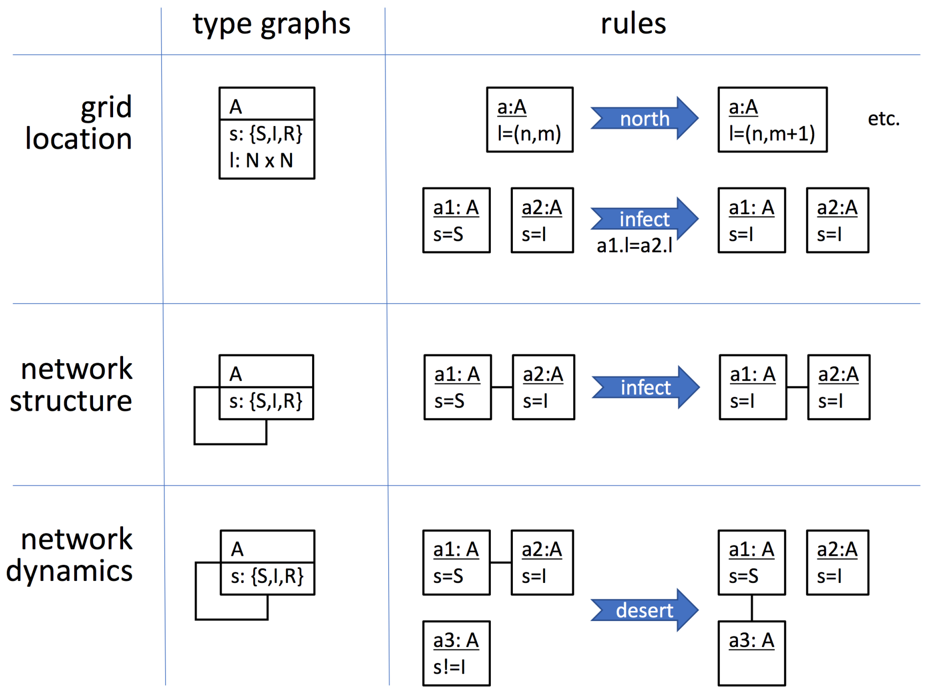

Figure 2: SIR model features

The basic SIR model may be able to predict levels infections in situations where agents have equal probability of interaction, but this is not in general the case. In order to get more accurate predictions we have to incorporate information about the conditions under which infections happen. Obvious factors are location and social connections. To account for these, the model can be extended as shown in Fig. 2. The *location* model adds a location attribute l to each agent and limits the *infect* rule to such cases where both agents are in the same location. In addition, rules are added for moving in different directions. Rule *north* is shown as an example. The *network* model extends the base model by allowing agents to be linked. There is no movement, but the *infect* rule is restricted to agents that are connected. In addition to network structure, we can introduce a *dynamics* as shown in the last model. A rule is added to allow agent $a1$ to switch connection, changing their behaviour to avoid infection through $a2$.

Apart from making predictions, agent-based models represent hypotheses about the mechanisms and factors affecting social processes. For example, we can compare predictions of different versions of the model with the observed behaviour to understand which of the conditions (location, static or changeable connections) are significant to the spread of certain diseases. It is worth stressing that ABMs used for prediction or as hypotheses about real processes will be much more sophisticated than the minimal examples chosen for this paper.

4 Feature Modelling and Composition

To support the use of feature modelling in ABMs, features need to be identified, specified, composed and assessed for their relevance. We also consider the extraction of reusable features into model libraries.

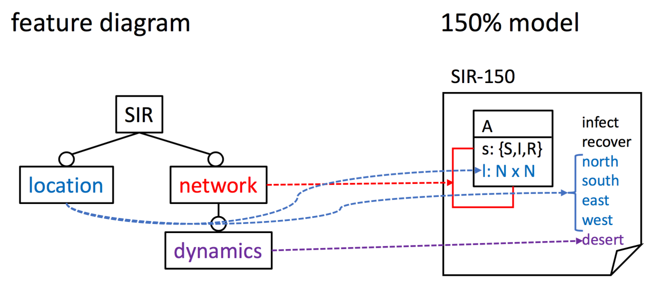

Figure 3: SIR feature model

Feature Identification For a given phenomenon (e.g., opinion formation or disease propagation) we want to know

- What are the features worth considering?
- What configurations of features are meaningful?

These questions are best answered studying existing literature or models, which is beyond the scope of this discussion. The result of such an analysis however can be expressed as a feature diagram $FD = (F,T)$ consisting of a set of features F and a tree-like diagram T over F as nodes describing valid configurations $C \subseteq F$.

Feature Specification and Composition. A feature model $FM = (FD,M,m)$ is made up of a feature diagram FD, an underlying model M incorporating all features, usually called the 150% model to indicate that it is not in itself a meaningful model but one requiring further restrictons, and a mapping m identifying features F in M. From this we can derive a variant MC of M for every valid configuration C.

The feature model for the SIR model and its extensions are shown in Fig 3. The tree-like diagram on the left shows that the base feature *SIR* can be extended by optional features *location* and *network* such that the latter admits a further extension by *dynamics*. The mapping of features to model elements (types and rules) is illustrated in the 150% model shown on the right. All unlabelled (black) elements belong to the base model. Elements in light blue, such as the l attribute and the movement rules, belong to the *location* feature, etc.

That means, M provides a combination of all possible features of the model, some of which may be mutually exclusive or redundant. There is no consideration for complex interaction of features at this stage, i.e., separate features, when added jointly, are assumed to be orthogonal.

The mapping of features to model elements provides an interpretation of the feature tree, where nodes represent graph transformation systems and edges are extension morphisms between systems. The semantic idea is that the behaviour of the extended model can be projected onto the smaller one, i.e., extension reflects behaviour.

Graph transformation systems can be composed along suitable morphisms. A configuration (set of features) defines a sub-tree of the feature diagram. If it exists, the colimit of this sub-diagram represents a system combining all features from this configuration [5].

Formally, each configuration C defines a graph transformation system GTS_C such that $C \subseteq D$ implies $GTS_C \subseteq GTS_D$. The inclusion is *conservative* if there are no new effects on old types. That means, if rules in the extended system are reduced to the base types, their effects coincide with those of the corresponding rules in the base system. In this case, behaviours in GTS_D can be projected onto behaviours of

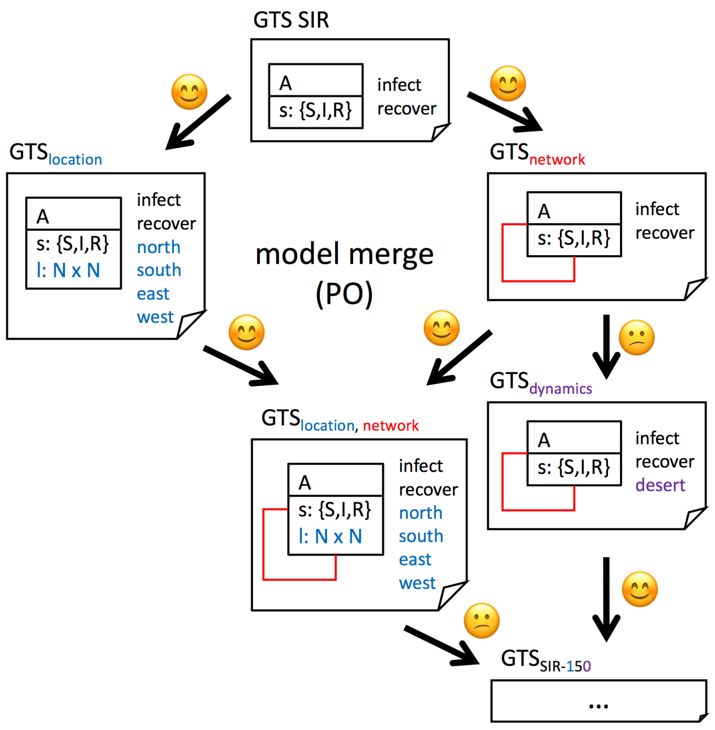

Figure 4: Variant models merging

GTS_C, i.e., GTS_C is a view of GTS_D. This is important because it allows us to compare two models in order to assess the relevance of their distinguishing features: If two models differing for a given feature do not show significantly different behaviour (as assessed, for example, by simulation), then the feature is not considered relevant to the behaviour of the model.

Fig. 4 shows how the systems representing the different extensions of the base model are composed. Orthogonal extensions of the type graphs are merged by adding both extensions independently to the new model, while rules are simply copied so the new model inherits all rules of both given models. It turns out that the extensions in the first merge operation are all conservative, but the extension of $GTS_{network}$ by $GTS_{dynamics}$ is not. The reason is that the *desert* rule creates a new effect on instances of the existing link type, so $GTS_{dynamics}$ behaviour cannot be mapped to that of $GTS_{network}$. However, $GTS_{dynamics}$ is a conservative extension of the basic *SIR* model.

5 Discussion and Future Work

We have outlined an approach to use graph transformation and feature models to create and organise agent-based models in the social sciences. So far we did not address the implementation of ABMs in a simulation language nor the question of calibrating a model's parameters and initial conditions, or validating its findings.

In order to judge if a given feature is relevant in the context of a model, we can compare model versions with or without an optional feature or containing one or another alternative feature. Based on the construction of the graph transformation systems corresponding to the relevant configurations, they are extensions of common base systems, and therefore their behaviours can be projected onto and compared from the perspective of this shared base. Combined with a suitable simulation approach this will allows us to compare results across different models and thus assess which feature configurations are relevant to understanding the phenomenon at hand.

The feature-oriented composition of systems is supported by tools such as FeatureMapper[1] which obtain a model configuration by filtering a 150 % model. A solution specifically for graph transformation systems using the Henshin tool[2] is the work on variability-based graph transformations [9] supporting optional or alternative structures within rules over the same type graph. While we follow a composition-based approach where each features are mapped to separate graph transformation system related by morphisms, the variability-based solution uses annotations on model elements to map features. The difference is mainly one of notation. In both cases a composed model such as $GTS_{location,network}$ would contain one rule which propagates an infection if the agents are both at the same location and related in the network.

Once features have been identified and assessed, we may want to reuse the components implementing them in other models. That means to abstract them from the 150% model they were originally embedded with by specifying the assumptions by which the components can operate as parts of other models and parameterising the components to make them more widely reusable. Component concepts for graph transformation models have been studied and may be applicable here, if they can be extended to stochastic or probabilistic models. In order to address the calibration of parameters and validation of models needed to link models to real data, notions of stochastic equivalence or similarity of models could be investigated.

A range of techniques exist to develop, analyse and maintain feature models in software engineering. For example, reverse engineering techniques can be used to support the extraction of feature models [8]. These are left to be explored for the usefulness in the case of agent-based modelling.

References

[1] Simon Angus & Behrooz Hassani-Mahmooei (2015): *"Anarchy" Reigns: A Quantitative Analysis of Agent-Based Modelling Publication Practices in JASSS, 2001-2012*. *Journal of Artificial Societies and Social Simulation* 18(4), p. 16, DOI: `10.18564/jasss.2952`. Available at `http://jasss.soc.surrey.ac.uk/18/4/16.html`.

[2] Edmund Chattoe-Brown (2013): *Why Sociology Should Use Agent Based Modelling*. *Sociological Research Online* 18(3), p. 3, DOI: `10.1177/0038038501035004006`. Available at `http://www.socresonline.org.uk/18/3/3.html`.

[3] Edmund Chattoe-Brown (2014): *Using Agent Based Modelling to Integrate Data on Attitude Change*. *Sociological Research Online* 19(1), p. 16, DOI: `10.1177/009365083010004004`. Available at `http://www.socresonline.org.uk/19/1/16.html`.

[4] Yen-Sheng Chiang (2013): *Cooperation Could Evolve in Complex Networks when Activated Conditionally on Network Characteristics*. *Journal of Artificial Societies and Social Simulation* 16(2), p. 6, DOI: `10.18564/jasss.2148`. Available at `http://jasss.soc.surrey.ac.uk/16/2/6.html`.

[5] Gregor Engels, Reiko Heckel, Gabriele Taentzer & Hartmut Ehrig (1997): *A Combined Reference Model- and View-Based Approach to System Specification*. *International Journal of Software Engineering and Knowledge Engineering* 7(4), pp. 457–477, DOI: `10.1142/S0218194097000266`.

[1]http://featuremapper.org

[2]https://www.eclipse.org/henshin/

[6] Segismundo S. Izquierdo, Luis R. Izquierdo & Nicholas M. Gotts (2008): *Reinforcement Learning Dynamics in Social Dilemmas*. *Journal of Artificial Societies and Social Simulation* 11(2), p. 1. Available at `http://jasss.soc.surrey.ac.uk/11/2/1.html`.

[7] Conrad Power: *A Spatial Agent-Based Model of N-Person Prisoner's Dilemma Cooperation in a Socio-Geographic Community*. *Journal of Artificial Societies and Social Simulation* 12(1), p. 8. Available at `http://jasss.soc.surrey.ac.uk/12/1/8.html`.

[8] Steven She, Rafael Lotufo, Thorsten Berger, Andrzej Wasowski & Krzysztof Czarnecki (2011): *Reverse engineering feature models*. In: *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu , HI, USA, May 21-28, 2011*, pp. 461–470, DOI: `10.1145/1985793.1985856`.

[9] Daniel Strüber, Julia Rubin, Thorsten Arendt, Marsha Chechik, Gabriele Taentzer & Jennifer Plöger (2017): *RuleMerger: Automatic Construction of Variability-Based Model Transformation Rules*. In: *Software Engineering 2017, Fachtagung des GI-Fachbereichs Softwaretechnik, 21.-24. Februar 2017, Hannover, Deutschland*, pp. 135–136, DOI: `10.1007/978-3-662-49665-7_8`.