Post-mortem Analysis of Emergent Behavior in Complex Simulation Models

Claudia Szabo
Department of Computer Science
The University of Adelaide
North Terrace
Adelaide 5005
claudia.szabo@adelaide.edu.au

Yong Meng Teo
Department of Computer Science
National University of Singapore
Computing 1, 13 Computing Drive
Singapore 117417
teoym@comp.nus.edu.sg

ABSTRACT

Analyzing and validating emergent behavior in component-based models is increasingly challenging as models grow in size and complexity. Despite increasing research interest, there is a lack of automated, formalized approaches to identify emergent behavior and its causes. As part of our integrated framework for understanding emergent behavior, we propose a post-mortem emergence analysis approach that identifies the causes of emergent behavior in terms of properties of the composed model and properties of the individual model components, and their interactions. In this paper, we detail the use of reconstructability analysis for post-mortem analysis of known emergent behavior. The two-step process first identifies model components that are most likely to have caused emergent behavior, and then analyzes their interaction. Our case study using small and large examples demonstrates the applicability of our approach.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*validation*; I/6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies*

Keywords

Emergent behavior, emergence, simulation, reconstructability analysis

1. INTRODUCTION

The behavior of complex systems cannot often be reduced only to the behavior of their individual components and systems require thorough analysis once unexpected properties have been observed [8, 17, 20]. These properties, called *emergent properties*, are increasingly becoming important as software systems grow in complexity, coupling, and geographic distribution [1, 15, 17, 20]. Examples of emergent properties include connection patterns in data extracted from social networks [6], trends in big data analytics [9], and power supply variation in smart grids due to provider competition [3]. More malign examples of emergent properties in computer systems include the Ethernet capture effect [24], router synchronization problems [10], and load-balancer failures in a multi-tiered distributed system [20]. Because emergent properties may have undesired and unpredictable effects and consequences, and unpredictable systems are less credible and difficult to manage, techniques for the identification and validation of emergent properties are becoming of crucial importance.

Despite ongoing research since the 1970s [1, 7, 11, 15, 26], very few methods for identifying, classifying, and explaining emergent properties exist [5, 18, 26, 27], and they are usually employed only on simplified examples that are not often found in real life [27]. These approaches can be classified broadly from two orthogonal perspectives, namely live analysis and post-mortem analysis [27]. *Live emergence analysis* proposes to identify emergence as it happens, using meta-models of calculated composed model states [27], or representations of system interaction [4]. *Post-mortem* emergence analysis starts from a definition of a known or observed emergent property and aims to identify its cause, in terms of model components and their interaction [5, 26].

A key challenge remains in the definition and identification of variables or attributes that describe the system sub-components, or the *micro-level*, and the system as a whole, or the *macro-level*, and the relationships between these two levels. Next, *weak emergence* [1, 15] results from the interaction between the model components at the micro-level but current approaches to identify weak emergence have only considered models with a small number of components. Current approaches [5, 18, 26] are demonstrated using simple models such as flocks of birds, and have limiting assumptions and constraints when applied to more complex systems. For example, most approaches do not consider mobile agents [18], or assume unfeasible a-priori specifications and definitions of emergent properties [27].

Simulation has been identified as a suitable means of analyzing complex systems [1], especially through the use of component-based models, that permit the study of increasingly large systems, both in size and complexity. To this extent, Page and Opper proposed a formal framework for analyzing the complexity of composition and the emergence of new behavior [22]. Other approaches compare well-known examples of emergence, such as simulations of flocks of birds, to simulations that exhibit no emergence at all, such as brownian movement, and establish interaction measures [4]. The emergence of a particular variable value can be identified in the simulation code using static and dynamic program analysis [14]. These approaches can be classified as post-mortem emergence analysis, because an emergent property is previously

SIGSIM-PADS'13, May 19–22, 2013, Montréal, Québec, Canada.

observed and defined, and the system is analyzed to identify its causes. Despite employing knowledge of the emergent behavior, the post-mortem analysis of emergence still has important issues [27]. Challenges lie in the lack of formalisms for the definition of the emergent property as a function of the composed model states and of the states of the model components of interest [27]. Moreover, existing approaches have only been applied to simplified models such as models of flocks of birds [5, 26] and predator-prey [26], with a small number of components. In our previous work, we have proposed an integrated approach for the identification and validation of emergence, looking at both live emergence analysis and post-mortem analysis [27]. Our previous work focussed on detailing live emergence analysis and how it could be applied to a simple model of flocks of birds.

This work enhances the discussion of post-mortem analysis and presents how it can be applied to more complex examples such as the game of life and traffic junction models. In this paper, we detail the use of reconstructability analysis (RA) [2] to facilitate post-mortem emergent behavior identification. Given a data model defined as a set of input/output variables and their values, reconstructability analysis can help to identify relations and interactions between input variables that cause particular output values. RA has considerable value for analyzing quantitative variables linked by non-linear relations [2], which makes it appealing for the study of emergent behavior as non-linearity has been shown to be one of the fundamental characteristics of complexity [7], and, inherently, of emergence [1]. The use of RA provides a simple and straightforward method to formalize the definition of micro and macro properties.

We propose an experimental study of post-mortem emergence analysis using examples of emergent behavior that have not been studied before in this context. The purpose of our case study is three fold. Firstly, current work analyzed the causes of emergent behavior using approaches similar to our proposed post-mortem analysis but only focusing on simple, small-scale examples such as flocks of birds or predator-prey models. Our case study starts from the game of life model and introduces larger models such as flocks of birds and traffic junctions. Secondly, while reconstructability analysis has been proposed to study complex systems, its suitability for understanding emergence has yet to be analyzed, in particular for systems with a large number of model components that are observed for a long period of time. Thirdly, while post-mortem analysis seems more straightforward than live emergence analysis, our examples show that post-mortem analysis is nevertheless a challenging task. For example, a key challenges is representing emergent behavior as a measurable macro property, and identifying the micro properties that might facilitate analysis.

This paper is organized as follows. We compare and contrast current work to the validation and identification of emergence in Section 2. Section 3 presents an overview of our proposed approach to the identification of emergence and discusses important concepts. Section 4 presents our experimental analysis, and Section 5 concludes this paper and presents future work.

2. RELATED WORK

An emergent property can be defined as "a property of an assemblage that could not be predicted by examining the components individually" [1]. Many characteristics of emergence have been identified in various works [1, 7, 15] and include: radical novelty (features not previously observed in systems); coherence or correlation (meaning integrated wholes that maintain themselves over some period of time); self-organization (individual model components organize into systems without pre-defined rules); a global or macro "level" (i.e. there is some property of "wholeness"); it is the product of a dynamical process (it evolves); and it is "ostensive" (it can be perceived); it is the result of rich interaction, but this interaction is short range (information is received primarily from neighbours). An important perspective for understanding emergence is the separation between *micro* and *macro* levels, that refer to abstraction at each individual level and at the level of the composed model as a whole respectively. Of interest to our approach is that the micro-level properties are usually measured by observing the component states, e.g. the collection of all variables and their values, of each system component. In contrast, the macro-level properties can be measured either as an aggregation of all the states of the system sub-components, or by observing the overall system behavior and trends.

In this paper we focus on *weak emergence* [1], a fundamental type of emergence that states that the properties of the whole, or the macro level, are resulted from the properties of the parts, or the micro level, and the interaction at the micro level, but that it is not trivial to infer the properties of the whole, and that extensive simulation studies are required to analyze and understand emergent behavior [15]. Approaches to identifying and understanding emergent behavior can be classified in three main categories, namely, *grammar-based*, *event-based* and *variable-based*, and further into live emergence analysis methods, where emergent behavior is not known beforehand, and post-mortem emergence analysis methods, that employ some specification or definition of emergence.

Grammar-based methods are live emergence analysis methods which aim to identify emergence in agent-based systems using two grammars, L_{WHOLE} to describe the properties of the system as a whole, and L_{PARTS} to describe the properties obtained from the reunion of the parts, and a definition of emergence as the difference between L_{WHOLE} and L_{PARTS} [18]. The behavior of an agent in a multi-agent system can be represented as a *grammar*, which is a set of rules that governs the formation of *words* using a set of *symbols*. L_{WHOLE} and L_{PARTS} can be easily calculated as sets of words that are constructed following agent behavior descriptions. This method does not require a prior observation of the system to identify possible emergent properties or behaviors, which makes it suitable for large-scale composed models where such observations are almost impossible. However, the nature of the formalism and the computation of the composed model states make it difficult to scale. *Event-based methods* are post-mortem analysis approaches in which behavior is defined as a series of simple and complex events that change the system state state [5]. Complex events are defined as compositions of simple, atomic events. Emergence is defined by a system expert as a complex event, and the approach focuses on determining the causes of emergence in terms of the sequence of complex and simple events in the system. In *variable-based methods*, a specific variable or metric is chosen to describe emergence. Changes in the values of this variable are said to signify the presence of emergence properties [26]. For example, the centre of mass of a bird flock could be used as an example of emergence in bird flocking behavior, as shown in [26]. The approach uses Granger causality to establish the relationships between a macro-variable, representing a system property, and micro-variables, representing properties of the system sub-components. This has the advantage of providing a clear process to identify emergence, that can be easily implemented. However, the approach requires system expert knowledge. Other approaches use metrics such as Shannon entropy [12, 23] and variety [16, 31] to measure emergence in a system. These approaches do not require system expert knowledge because they use general definitions rooted in complex systems the-

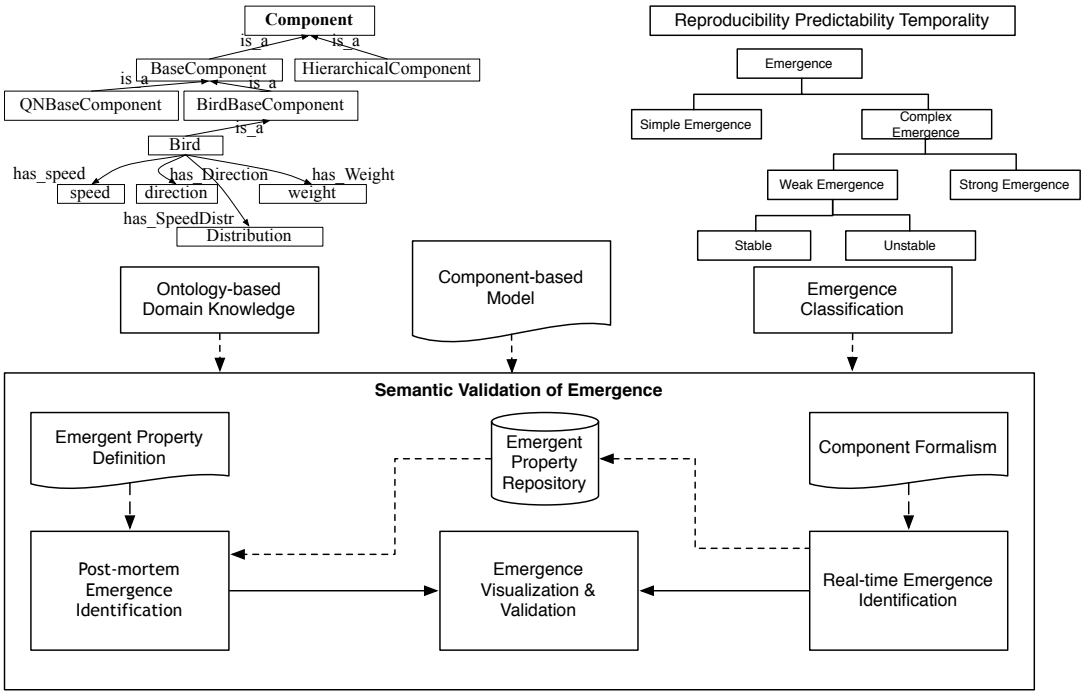

Figure 1: Integrating Orthogonal Perspectives in Emergence Validation

ory. However, to date they have only been applied to simple toy examples.

Page and Opper [22] propose a formal framework for analyzing the complexity of composition and emergence. They propose a definition emergent behavior consisting of the components a and b i a composition $a \diamond b$, an objective o, and the "satisfies" operator $\models$. If $a \models o$, then a satisfies objective o. If $a \not\models o$ and $b \not\models o$, but $(a \diamond b) \models o$, then we can say that the composition is *emergent*. Similar to variable-based methods, Gore and Reynolds denote emergence as a specific variable value and propose to highlight the lines in the simulation source code that cause that particular value [14]. They further propose a taxonomy for analyzing emergent behavior based on reproducibility, predictability, and temporality [13], but the identification of emergent behavior as well as its validation are not addressed. Another measure of emergence is the interaction between agents in an agent-based model [4], which is defined as an agent-specific counter that increases as the agent interacts in terms of direct message passing with other agents in the environment. Emergence is said to appear if the interaction measure deviates from normality. This approach provides a straightforward measure of emergence. However, cases where emergence is a result of indirect interaction between agents, such as in flocks of birds [25] and in Conway's Game of Life [11] are not addressed.

3. ANALYZING EMERGENT BEHAVIOR

This section presents an overview of our post-mortem emergence analysis within our proposed integrated framework for the identification and validation of emergent behavior. We present an overview of our framework, discuss the steps for our proposed post-mortem analysis and highlight the use of reconstructability analysis within our approach.

3.1 Overview

In our previous work [27], we proposed an integrated approach for the identification and validation of emergent behavior. As shown in Figure 1, we identify two orthogonal perspectives to understand emergent properties, namely, live and post-mortem. Firstly, *live emergence analysis* identifies emergence as it happens, without requiring prior knowledge of emergent properties. The approach relies on a representation of model components in terms of *what* they achieve rather than *how*, using an objective-based meta model component. A composed model state, or the *macro* is constructed from the states of its model components. The constructed state is then compared with the observed simulation state and significant differences are highlighted to the system expert. Live emergence relies on an ontology-based representation of specific domain knowledge represented in the figure as COSMO, our proposed ontology for component-based simulation [29]. Subsequently, properties identified as emergent are saved into the emergent property repository. Despite its appeal, live emergence analysis requires model components to be specified formally in terms of their objective, which may be difficult to obtain in practice.

Secondly, if emergent behaviors are identified beforehand, *post-mortem emergence analysis* can be applied to cause in terms of interaction between model components, their attributes and values. This assumes that emergent behaviors have been identified at the macro or system level, and that the model component properties, or the micro properties, can be measured. The information obtained from the analysis is saved into the emergent property repository, which is a collection of properly specified and defined emergence properties that leverage on existing work in the classification of emergence with respect to type, application domain, and specific occurrence among others [1, 13].

Arguably the more straightforward of the two perspectives, post-mortem emergence analysis remains challenging. Firstly, as discussed above, existing approaches have studied models with a small number of components and a low level of detail in the component representation. This affects the number of micro properties that can

be considered in the analysis. Secondly, it is difficult to formally define the observed emergent behavior as a system-wide property at the macro level and a collection of component attributes at a micro level. Next, defining a formal process for identifying the relationships between micro and macro levels in a manner that is independent from the system under study to facilitate automated emergence identification remains a key challenge. Towards this, we propose the use of reconstructability analysis (RA) [2, 32] for the post-mortem analysis of emergent behavior. RA facilitates the study of systems with a large number of micro properties and can help to identify specific interactions between model components that lead to the emergence of a macro property.

3.2 Post-mortem Emergence Analysis

Post-mortem emergence analysis aims to identify the causes that lead to the identified emergent properties. Our approach relies on a representation of model components that captures component attributes and behavior. More formally, model components are represented by *meta-components*, $C_i = \langle R, A_i, B_i \rangle$, which describe the component *required attributes* R, *specific attributes*, A, and *behavior*, B. The required attributes are common to all components and are generally employed for version control, e.g. `author`, `location`, `lastUsed`. Examples of specific attributes include `interArrivalTime`, `speed`, `direction` etc. Component behavior describes the data that it receives and outputs as a set of states. The transitions between states are defined as a set of triggers expressed in terms of input, time and conditions.

Post-mortem emergence analysis is divided in three main steps as shown in Figure 2. Firstly, an emergent property, EP, for a

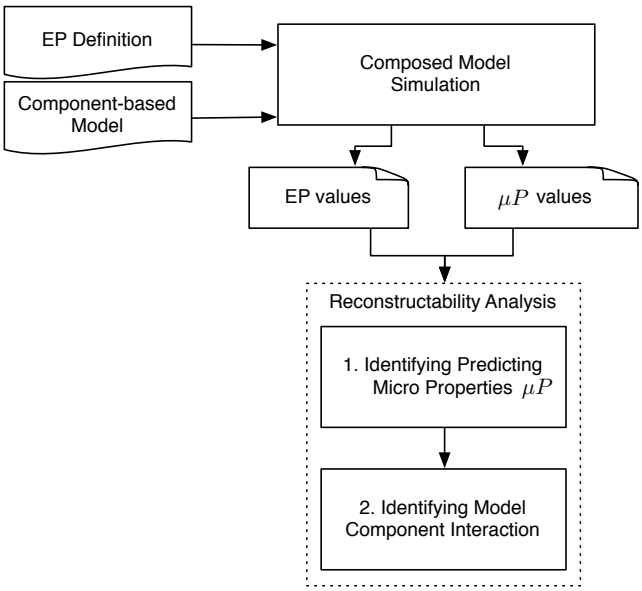

Figure 2: Post-mortem Emergence Analysis

composed model is defined based on previous knowledge extracted from the system expert or from observations of the system. Secondly, to identify the causes of EP, the composed model simulation is executed and the model components, representing the *micro* level, are observed throughout the simulation run and recorded as a set of values μP. The values of EP, or the *macro* level, are also recorded. Thirdly, we employ reconstructability analysis to determine the relationship between μP and EP. For a set of recorded values for the micro properties $\mu P_{i\,j}$ at every simulation time step j, and the set of recorded macro properties EP_j, we construct a system in which the inputs are μP_i and the output is EP, and the observations are $\mu P_{i\,j}$ and EP_j at every simulation step j. We divide our application of RA into two steps, namely, the identification of predicting micro properties $\mu P_{predict}$ that are most probable to generate EP, and the detection of interaction between model components.

3.2.1 Reconstructability Analysis

RA decomposes a system expressed as data in the form of set theoretic relations or multivariate probability distributions, into parts, namely relations or distributions involving subsets of variables [2, 32]. Of interest to our approach is the fact that observational data can be modeled and compressed by variable-based decomposition. RA models, which specify the inter-dependencies and interactions among the variables, are selected to minimize error and complexity. RA has considerable value for analyzing quantitative variables linked by non-linear relations [2], which, as discussed above, makes it appealing for studying emergent behavior where non-linearity is frequent [1, 7].

Two types of systems can be analyzed using RA, namely, direct and neutral [32]. In a *direct* system, a number of input variables are analyzed for their capability of predicting one or more output variables. In a *neutral* system, there is no distinction between input and output variables. Given a system represented by a number of variables, a reconstructability analysis tool such as Occam3 [32, 21] searches through a lattice of possible model component or model inter-dependencies structures. These structures are analyzed based on observed data, and goodness measures, such as information entropy, type 1 errors, structure complexity, model confidence, etc. are calculated.

For example, given three variables A, B, C the possible structures to be considered for a directed system are shown in grey in Figure 3 and in white for neutral systems (Figure adapted from [32]). In a directed system, for inputs A, B, and output C, five model structures will be considered, namely, ABC, AB:AC:BC, AB:AC, AB:BC, AB:C. In model ABC, A and B interact in their joint effect on C. Structure AB:C signifies that the output is independent of the input. Structure AB:AC signifies that variable A best predicts output variable C, and structure AB:AC:BC signifies that output variable C depends separately on A and on C, and so on. Structure A:B:C is called an independence model for the neutral system. For a directed system with output C, the independence model is AB:C. definition of a parent-child relationship. Given a particular structure, each child structure is created by removal of a relation and reinsertion of all the embedded relations within that relation which are not already present in other relations of the structure. A search through this lattice analyzes each model based on the measures defined above.

Reconstructability analysis promises to scale well in terms of the number of properties and their observations, with RA tools such as Occam3 being proven to analyze systems with up to 2^{16} observations of micro variables [32]. Moreover, in contrast to existing work, it also permits the formal and statistical analysis of the influence of component interaction on the emerging behavior, which increases the insight into the system execution.

In this paper, we employ the Occam3 reconstructability analysis tool proposed by the Systems Science group at the Portland State University. Occam3 [21] provides a web-based interface that allows users to upload a data file with variable observations, performs the required analysis, and permits the viewing and downloading of results. For directed systems, Occam3 reports measures such as Shannon entropy, the probability of Type 1 errors, and the uncertainty reduction. Alpha (α), the probability of making a Type 1

Model	Problem Size	Micro Property (μP) & Macro Property (EP)
Game of life	10 x 10 grid size	μP: $state(i,j)$ - state of cell (i,j): alive/dead EP: Δe - rate of expansion of alive cells
Flock of birds	10 / 20 / 50 birds	μP: $d(b_i, CP)$ - distance from *bird i* to the center of the drawing panel EP: $d(CM, CP)$ - distance from center of mass of flock to the center of the drawing panel
Traffic junction	10 / 50 / 100 cars	μP: v_i - velocity of car i EP: Δc - rate of growth of traffic congestion
		μP: p_i - politeness of car i EP: Δc - rate of growth of traffic congestion
		μP: (v_i, p_i) - velocity and politeness of car i EP: Δc - rate of growth of traffic congestion

Table 1: Experiments Overview

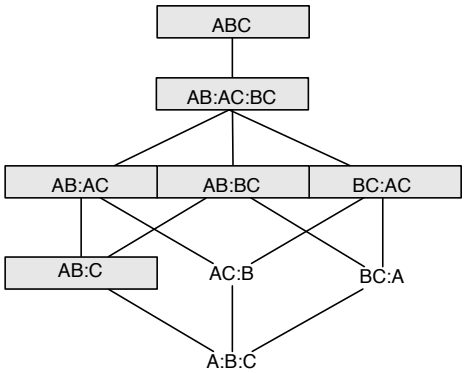

Figure 3: Lattice of Specific Structures for a 3-variable System

error, is the probability of being in error if one rejects the null hypothesis that the model is the same as the reference model. The uncertainty reduction refers to the percent reduction in the uncertainty of the output variable given the independent variables in the predicting components. When employing reconstructability analysis for directed systems, a model is considered good if it has a high information content, a small complexity (i.e., a small number of structures), and a low α.

3.2.2 Post-mortem Analysis using RA

Reconstructability Analysis is helpful in the study of complex systems because it can determine the model components and their interaction that best describe the system. In post-mortem analysis, the system expert highlights the *macro* property that best characterizes the emergent property, and thus in some sense there is some understanding of what component attributes, i.e., *micro* properties, cause it. With the help of RA, we propose to determine the model components whose interaction, direct or indirect, with the greatest influence on the emergent property. Furthermore, we aim to determine the component interaction that best causes emergence. In the following, we view the composed model as a *directed* system from the RA perspective, with the output being the macro property, and the input being the observer values of the micro properties, i.e., the component attributes and their values. Using the Occam3 reconstructability analysis tool, we follow a two-step process to determine model component interaction that causes emergent behavior, as shown below.

Step 1. Identifying Predicting Micro Properties
In the first step, the model components that are most likely to cause the emergent property EP are identified. This is done by performing an upwards search in the lattice using the independence model containing all micro properties as reference. We stop the search when adding a new independent variable to the search is not statistically significant, and construct the set of predicting micro properties $\{\mu P_{predict}\} \subseteq \{\mu P_i\}$ For this upward search, we are looking for models with low α. For example, if four micro properties and one macro property exist, namely, A, B, C, D, and Z respectively, the independence model is $ABCD$:Z, which says that each micro property can predict the macro property equally.

Step 2. Identifying Model Component Interaction
In the second step, we aim to determine the interaction between the micro properties in $\mu P_{predict}$ that is most significant to determine EP. This is done by performing an upwards search in the lattice using a disjoint reference model, where components are formed from the predicting micro properties $\mu P_{predict}$. For example, if two predicting micro properties A, B have been identified in step 1 from the set $\{A, B, C, D\}$, the search would start from the disjoint reference model AB:AZ:BZ. The result would indicate interaction between A and B if alpha is small for the model ABZ and if it reduces the uncertainty of Z by more than the reference model. For three variables A, B, C, and output Z, the reference model ABC:ABZ:ACZ:BCZ is better than ABC:AZ:BZ:CZ because it ascertains that the interaction involves all three variables.

4. EXPERIMENTAL ANALYSIS

Our experimental analysis considers a case study of post-mortem emergence analysis of three composed models, namely, the game of life, flocks of birds, and traffic junctions. To better highlight the suitability of our approach and reconstructability analysis, our experiments present small and large models, with respect to the number of components and the number of attributes included in the analysis, as summarized in Table 1. For the game of life model, we attempt to use a previously published definition of a macro property, namely, the rate of expansion of live cells, but find it lacking for the particular example under study, and as such propose a new definition. On the other hand, using our proposed approach, we were able to confirm annecdotal and CCTV-camera studies [19] that suggest that driver politeness causes traffic jam. Moreover, we study models for which the micro property, μP, is expressed in terms of more than one attribute, as in the case of the traffic junction model.

We implemented the flocks of birds and game of life models, and the traffic junction model is adapted from a microsimulation of road traffic flow [30]. The source code includes observations of micro and macro properties and the output is in a format suitable for our chosen reconstructability analysis tool. We employ the web-based implementation of Occam3 [32]. Experiments are executed on a 2.7GHz Intel Core i5, 16GB RAM desktop machine, and, except

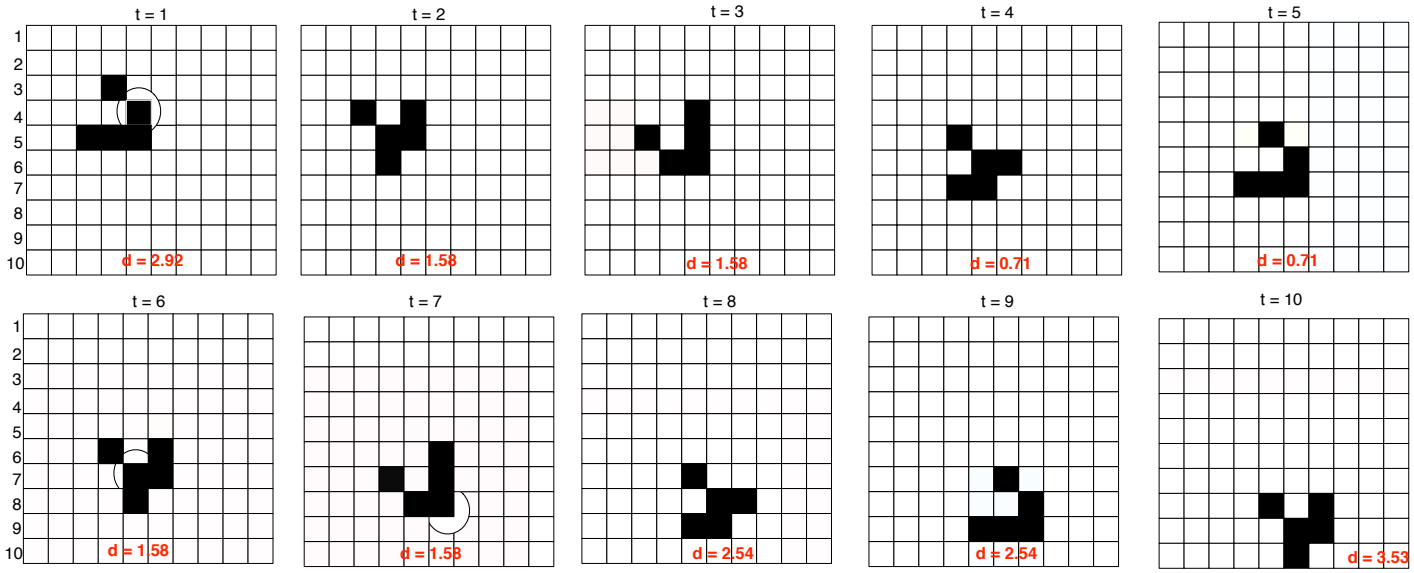

Figure 4: Glider in the Game of Life

for the game of life implementation, the simulation output values represent averages of 10 runs.

4.1 Game of Life Model

The Game of Life is a well-known cellular automaton in which the world is represented as a two-dimensional grid of cells and each cell has two possible states, alive or dead [11]. Every cell interacts with its neighbors and at each step, the cells may change state from alive to dead and vice-versa according to the game of life rules [11]:

- *under-population*: a live cell with fewer than two live neighbours will die
- *normal life*: a live cell with two or three live neighbours remains alive
- *over-crowding*: a live cell with more than three live neighbours will die
- *reproduction*: a dead cell with exactly three live neighbours will become a live cell

An initial pattern represents the seed of the system. In our experiment we look at the well-known glider pattern, which is a moving, recurring pattern of an spaceship-like arrow, as shown in Figure 4. In our experiment, we abstract the cell as a model component with two attributes, `position` and `state`. Our automaton is executed for the first ten steps, data is collected and an input file for Occam3 is generated. At each step, we record the state of each cell i (alive or dead) as the micro property μP_i. In the literature, the rate of expansion of live cells has been suggested as a possible macro property for the Game of Life [1, 7]. This is not however possible in the case of the glider pattern, because the number of alive cells remains constant. As such, we record the distance between the center of the smallest bounding box that surrounds the glider and the center of the grid as the macro property EP. These distance values are shown in Figure 4. A part of the input file is shown in Figure 5. Each data row has 101 values, representing the cell states (μP_i) and the distance (EP) respectively, with the last value representing the distance.

An Occam3 input file first starts with the definition of the subcomponent variables, which are defined as `name, cardinality, input/output, short_name`. For example, for variable `Cell1`, representing the first top-left cell in the grid, there are two possible values (alive or dead), hence a cardinality of 2, and the variable is an input value, hence value 1. Observations about the values of the specified variables follow after the `data:` token. Because the size of the input data is large, we show in Figure 5 the alive cells as line 5 in the grid for the first five observations, and line 7 in the grid for the last five observations.

```
:nominal                :data
Cell1,2,1,Cell1        0 0 0 ...0 0 1 1 1 0 0 0 0 0 ...4
Cell2,2,1,Cell2        0 0 0 ...0 0 0 1 1 0 0 0 0 0 ...2
Cell3,2,1,Cell3        0 0 0 ...0 0 1 0 1 0 0 0 0 0 ...2
Cell4,2,1,Cell4        0 0 0 ...0 0 0 1 0 0 0 0 0 0 ...1
Cell5,2,1,Cell5        0 0 0 ...0 0 0 0 1 0 0 0 0 0 ...1
Cell6,2,1,Cell6        0 0 0 ...0 0 0 0 1 0 0 0 0 0 ...2
Cell7,2,1,Cell7        0 0 0 ...0 0 0 0 1 1 0 0 0 0 ...2
Cell8,2,1,Cell8        0 0 0 ...0 0 0 0 0 1 1 0 0 0 ...3
Cell9,2,1,Cell9        0 0 0 ...0 0 0 0 0 0 1 0 0 0 ...3
Cell10,2,1,Cell10     0 0 0 ...0 0 0 0 1 0 1 0 0 0 ...5
...
D,5,2,D
```

Figure 5: Occam3 Input File for the Game of Life Model

Step 1. Identifying Predicting Micro Properties

As discussed above, the first step in our method identifies model components that are most likely to cause the emergent property EP by performing an upwards search in the lattice using the independence model containing all micro properties μP_i as the reference. The cells are numbered starting with cell one, in the top left corner of the grid, and proceeding from left to right until cell 100, in the bottom right of the grid. The independence model for the game of life model is

$$Cell1 Cell2 \ldots Cell100 : D \text{ or } IV : D$$

where IV means *independed variables*. Table 2 presents the results. As discussed above, a good set of predicting micro proper-

Model	Entropy	Complexity	α
IV : D	5.45	0	0.00
IV : Cell65Cell77D	4.13	12	0.11
IV : Cell75Cell77D	4.09	12	< 0.01
IV : Cell75Cell86D	4.09	12	< 0.01
IV : Cell56Cell75Cell77D	3.59	28	0.59
IV : Cell56Cell75Cell86D	3.59	28	0.59
IV : Cell35Cell75Cell86D	3.59	28	0.59

Table 2: Predicting Micro-Properties using RA

ties achieves in a reconstructability analysis search a small value for the measure of complexity in terms of the number of structures, and a low α. From the results in Table 2 it can be seen that micro properties *Cell*75 and *Cell*86, highlighted in Figure 4 using a circle, are the best predictors and generators for the macro property *D*. They are the cells that stay alive and interact the most in the second glider pattern that is formed from iteration 6 to 9. In a similar manner, *Cell*75 and *Cell*77 are equally good candidates. It is important to highlight here that, if a larger number of predicting micro properties were to be considered, the set containing *Cell*35, *Cell*75, and *Cell*86 would also be a good candidate. *Cell*35 is also highlighted in Figure 4, and represents one of the cells in the first glider pattern that remains alive the longest. We define set $\mu P_{predict}$ as $\mu P_{predict} = \{Cell75, Cell86\}$.

Step 2. Identifying Model Component Interaction
The next step proposes to determine if the interaction between the micro properties in set $\mu P_{predict}$ is the cause of the values of the macro property *EP*. The reconstructability search starts from a disjoint reference model that is constructed from the micro properties in $\mu P_{predict} = \{Cell75, Cell86\}$ as

$$IV : Cell75Cell86 : Cell75D : Cell86D$$

The results in Table 3 show that *Cell*75*Cell*86*D* has an $\alpha = 0.00$ and reduces the uncertainty of output macro variable *D* by more than the disjoint reference model. This analysis shows that the

Model	Entropy	Complexity	α	Uncertainty Reduction (%)
Cell75Cell86D	2.13	4	0.00	64.60
IV : Cell75D : Cell86D	19.13	0	1.0	56.23

Table 3: Identifying Interaction in the Game of Life

interaction between the micro properties of *Cell*75 and *Cell*86 is responsible for the emergence of property *EP* represented by the distance between the center of the bounding box that surrounds the glider and the center of the grid. Using reconstructability analysis, we are able to identify indirect interactions and the influences between micro properties. This would not be possible using current direct measures of interaction.

4.2 Flocks of Birds Model

Our flock of birds experiment looks at the well-known boid model [25], in which each component abstracts a moving bird whose position changes based on a set of simple rules that define its current position and the position of other birds in the flock. These rules are

- *separation* - individual bird steer to avoid crowding the other birds in the flock
- *alignment* - individual steer towards the average herding of local flock mates
- *cohesion* - individual moves towards the average position of local flock mates

The boid model has been shown to exhibit emergent behavior of flocking and of flocking after encountering an obstacle, when the flock splits and reunites.

In our experiment we abstract a bird as a model component with `position` and `speed` as attributes, among others.We define the emergent macro property as the trajectory of the center of mass (CM) of the flock of birds. This is in accordance with previous studies that looked as emergence in the flock of birds model [26].

The simulation of the flock of birds is executed, and data is collected, pre-processed, and collated into a file that is then supplied to Occam3. As an example we employ ten birds that go through a sequence of 20 position changes according to the rules specified above, starting from random positions on the drawing panel. At each simulation step, each bird changes position according to the rules specified above; we log each bird position, as well as that of the center of mass of the flock. We reduce the number of variables by computing the distance d_i from the center of the environment, i.e. the center of the drawing panel. The values of d_i are saved into an output file, similar to that shown in Table 4 for the first ten steps.

Bird 1	Bird 2	Bird 3	Bird 4	Bird 5	Bird 6	Bird 7	Bird 8	Bird 9	Bird 10	CM
53	209	128	67	43	104	121	130	118	95	69
45	174	112	65	44	91	104	111	103	81	65
40	129	88	60	43	72	81	84	80	61	61
39	76	59	51	38	49	52	53	55	38	62
38	27	42	34	37	31	19	30	31	23	69
54	32	48	11	57	9	31	53	42	36	80
59	71	69	13	75	18	70	102	81	69	101
56	113	100	42	94	51	111	153	125	109	131
58	154	136	79	118	92	155	202	171	153	169
82	195	175	125	151	140	200	247	213	202	213

Table 4: Distance from the Center of Drawing Panel

To further reduce the number and values of the variables, we discretize the distance values, as $floor(d_i/100) + 1$. The results are then transformed into an Occam3 input file, as shown in Figure 6.

```
:nominal     :data
b1,4,1,a     1 3 2 1 1 2 2 2 2 1 1
b2,4,1,b     1 2 2 1 1 1 2 2 2 1 1
b3,4,1,c     1 2 1 1 1 1 1 1 1 1 1
b4,4,1,d     1 1 1 1 1 1 1 1 1 1 1
b5,4,1,e     1 1 1 1 1 1 1 1 1 1 1
b6,4,1,f     1 1 1 1 1 1 1 1 1 1 1
b7,4,1,g     1 1 1 1 1 1 1 2 1 1 2
b8,4,1,h     1 2 1 1 1 1 2 2 2 2 2
b9,4,1,i     1 2 2 1 2 1 2 3 2 2 2
b10,5,1,j    1 2 2 2 2 2 3 3 3 3 3
cm,4,2,k       ...
```

Figure 6: Occam3 Input File for the Boid Model

Step 1. Identifying Predicting Micro Properties
The first step identifies the model components that are most likely to cause the emergent property *EP* by performing an upwards search in the lattice using the independence model containing all micro properties μP_i as the reference. The independence model for the boid model observed above is

$$ABCDEDFGHIJ : K \text{ or } IV : K$$

Table 5 shows the results of the search. As discussed above, we

Model	Entropy	Complexity	α
IV : K	5.85	0	1.00
IV : HK	4.83	9	< 0.01
IV : GK	4.78	9	< 0.01
IV : FGK	4.02	45	0.68
IV : GJK	4.36	57	0.94
IV : GHK	4.32	45	0.58
IV : GHJK	4.18	237	0.46

Table 5: Predicting Micro-Properties using RA for the Game of Life

are looking for a structure from a set of predicting micro properties that has a small value for the measure of complexity in terms of the number of structures, and a low alpha. The results presented in Table 5 show that micro properties F, G, H are the best predictors and generators for the macro property K. As such, the set $\mu P_{predict}$ is defined as $\mu P_{predict} = \{b7, b8, b9\}$. It is important to highlight here that sets $\{b7, b8\}$, $\{b7, b9\}$ and $\{b5, b6\}$ are also good candidates for $\mu P_{predict}$, but include a smaller number of micro properties.

Step 2. Identifying Model Component Interaction
We next attempt to determine if the interaction between the micro properties in $\mu P_{predict}$ is the cause of the values of the macro property EP. As discussed above, we start the reconstructability search from a disjoint reference model. For set $\mu P_{predict} = \{b7, b8, b9\}$, we use the disjoint reference model

$$IV : GHJ : GHK : GJK : HJK$$

As shown in Table 6, model $GHJK$ has an $\alpha = 0.00$ and reduces the uncertainty of output macro variable K by more than the disjoint reference model.

Model	Entropy	Complexity	α	Uncertainty Reduction (%)
GHJK	3.48	108	0.00	72.12
IV : GHJ : GHK :GJK:HJK	17.48	0	1.0	61.24

Table 6: Identifying Interaction using RA

In a similar manner, we execute post-mortem analysis for a flock of birds model with 20 and 50 individuals respectively. A summary of the results is shown in Table 10. As it can be seen, a large number of individuals interact to generate EP for the model of size 50. We believe that a reason for this is that the initial condition of the simulation caused by the size of the drawing panel, causes the individuals to be in an already flocked initial position as shown in Figure 7.

4.3 Traffic Junction Model

In this experiment we employ a traffic simulation model adapted from [30]. For simplicity, we simulate only two types of vehicles, cars and trucks, that are distinguished based on their speed. Another parameter of interest is driver politeness, which reflects an intelligent driver model that allows other vehicles to change lanes ahead of it. In our analysis, we employ models of 10, 50, and 100 vehicles respectively, and at every time step record the driver speed as a micro property, and the size of the congested area as a macro property. A congested area is shown in red in Figure 8 on the northwest side of the ring road. An additional micro property recorded is driver politeness. Recent studies of driver behavior used CCTV camera videos from a UK highway [19] to prove that driver politeness is the cause of the emergence of traffic jams. We arrive at the same conclusion using reconstructability analysis as shown below.

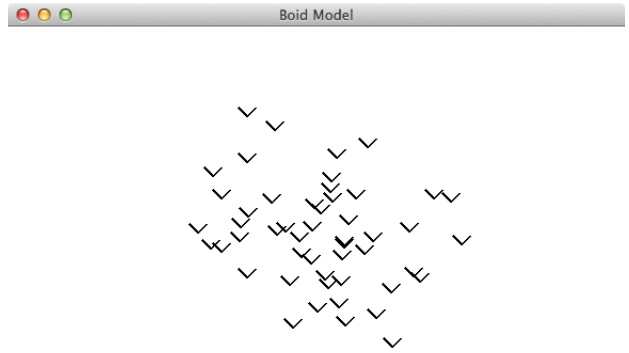

Figure 7: Flock of Birds Model

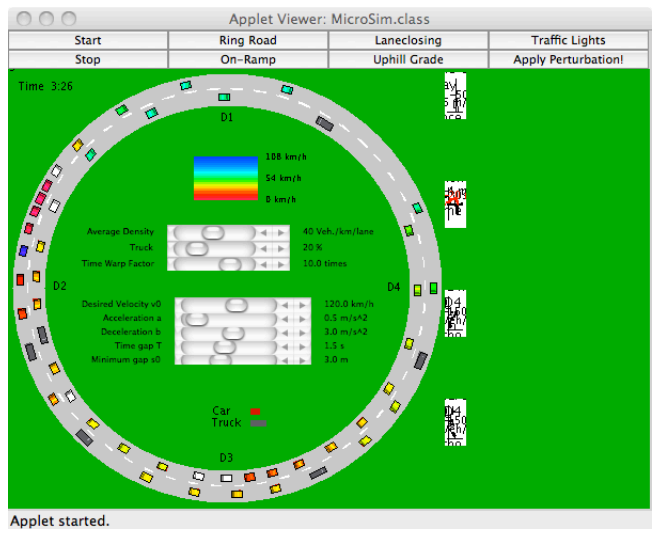

Figure 8: Emergence of Traffic Jam

A model with ten vehicles exhibits no emergence of traffic jam as the number of vehicles in the ring road is too small for the traffic to become congested. For a model with 50 cars, we introduce three impolite drivers, driving vehicles with id `Car4`, `Car16`, and `Car23` respectively. Our first experiment records vehicle speeds as the micro property. In a similar manner to the flocks of birds model, we execute the simulation for two minutes and record vehicle speed and size of the congested area. In all experiments, the drivers of the above vehicles are polite for the 30 seconds, impolite for a minute, and revert to politeness for the last 30 seconds. An analysis of the data shown in Table 7 reveals that no predicting variables can be identified, and that the independence model is the most trustworthy for these observations.

In the second experiment we record the politeness of each driver as a micro property.

Step 1. Identifying Predicting Micro Properties
As it can be seen in Table 8, the vehicles with impolite drivers are identified as predicting micro-properties for the macro property

Model	Entropy	Complexity	α
IV : Car28Car6Road	10.02	11594	1.00
IV : Car28Road	11.41	612	1.00
IV : Car24Road	11.43	612	1.00
IV : Road	13.57	0	0.00

Table 7: Vehicle Speed as Prediction for Congestion

`Road`. It is important to notice here that all impolite drivers have

Model	Entropy	Complexity	α
IV : Car0Car1Car23Road	4.74	31	< 0.01
IV : Car0Car23Road	4.74	31	< 0.01
IV : Car10Car16Road	4.74	31	< 0.01
IV : Car10Car23Road	4.74	31	< 0.01
IV : Car10Car4Road	4.74	31	< 0.01
IV : Car4Road	4.74	31	< 0.01
IV : Car16Road	4.74	31	< 0.01
IV : Car23Road	4.74	31	< 0.01

Table 8: Predicting Micro-Properties for Impolite Drivers

been identified as individually predicting the traffic congestion, and that all structures with up to three components are equivalent in terms of complexity and entropy. This is also the case when the drivers are impolite for the entire duration of the simulation run. Looking for a larger structure in terms of the micro properties, we define $\mu P_{predict}$ as $\mu P_{predict} = \{Car0, Car1, Car23\}$.

Step 2. Identifying Model Component Interaction
We analyze whether the interaction between impolite drivers and the other drivers on the ring road can be captured by RA. Towards this, we look at the disjoint model

$$Car0Car1Car23 : Car0Car1Road : Car0Car23Road : Car1Car23Road$$

As shown in Table 9, model $Car0Car1Car23Road$ has an $\alpha = 0.00$ and reduces the uncertainty of output macro variable *Road* by more than the disjoint reference model, as shown by the last column in the table.

Model	Entropy	α	Uncertainty Reduction (%)
Car0Car1Car23Road	4.74	0.00	28.75
IV:Car0Car1Car23:Car0Car1Road	6.74	1.0	20.56

Table 9: Identifying Driver Interaction

In our third experiment, we combine the two micro properties, velocity and politeness, into a single variable (v_i, p_i) defined as their product, $v_i * p_i$. The results are similar to those presented above, in that the interaction between impolite drivers and other cars are deemed to be the cause of emerging traffic jams.

4.4 Discussion

Table 10 summarizes our key results. Our runtime analysis highlights that our approach scales well even for large models. This is important when considering that other emergence identification approaches proposed in the literature can only be applied to small models [28]. Our proposed post-mortem analysis highlights to the system experts the entities whose interaction is most likely to generate particular behaviors. This information can be used to improve future simulation runs in two ways. Firstly, interactions between entities that are similar to those that have generated emergent behaviors can be highlighted to the user, to increase insight into the simulation results. Secondly, an analysis of these interactions can be used in the model validation.

4.4.1 Micro and Macro Properties

Our experimental analysis also shows that the success of our approach depends on how well the micro and the macro properties are chosen. For example, in our analysis of the game of life model, we did not use the *EP* suggested by literature, i.e., the rate of expansion of live cells because it was not appropriate to the glider pattern under study. Instead, we employed a definition of *EP* that captured the semantics of the glider, i.e., its movement, by looking at the distance between the surrounding bounding box of the glider and the center of the grid. Moreover, our analysis of the traffic junction model confirmed that, counter-intuitively, the driver speed cannot be shown to be the cause of the emergence of traffic jams, whereas driver politeness has been shown to cause traffic jams. These observations only show that post-mortem analysis, despite being more straightforward than live emergence analysis, remains challenging.

4.4.2 Scalability

Our experiments show that our approach can be applied to large-scale models. Occam3, the tool we employ for reconstructability analysis, can analyze very large state spaces with up to 2^{16} states, and as such the scalability of our approach is limited to models for which the state space generated by the model entities and their attributes does not exceed this size. For the traffic junction model, we have applied reconstructability analysis to models of up to 500 entities. From a practical perspective, difficulties arise from the specification of micro and macro properties, as a large and complex model is difficult to observe, and some micro or macro properties might be missed by the system expert. Towards this, a visualization tool that highlights behaviors and systems states that might be of interest to the system expert is needed.

4.4.3 Multi-dimensional Variables

Besides the main challenge in identifying system-specific definitions of micro and macro properties to facilitate analysis, another important issue remains the analysis of multi-dimensional micro-properties. For example, in the traffic junction model, while we have shown that the driver politeness causes the emergence of traffic jams, more insight could be obtained by analyzing the impact of driver politeness and driver speed together on the emergence of traffic jams. A limitation of the RA tool used is that it can only analyze single-dimension variables. Formulae can be devised to transform these micro-properties into single-dimension variables, but with a loss of meaning in the analysis.

5. CONCLUSION

Identifying and analyzing the causes of emergent behavior remains a challenging task even when system expert knowledge about emergence is available. In this paper, we propose a two-step approach for the post-mortem analysis of the causes of known emergence, in terms of model component attributes and values, as well as model component interaction. Firstly, we propose to use reconstructability analysis (RA) to identify micro properties that contribute to the cause of emergent macro properties identified by a system expert. Secondly, from the set of predicting micro properties, we identify using RA if the interaction between the respective model components causes the emergent property.

In contrast with current approaches, we apply our approach to both small and large examples defined in terms of the number of model components in the model, as well as the number of micro properties considered. Our experimental analysis of the game of

Model	Size	EP	μP	EP Identified	EP causes	Runtime (s)
Game of life - glider	10 x 10	Δe	$state(i,j)$	Yes	Indirect interaction between two cells	4.6
Flock of birds	10	$d(CM,CP)$	$d(b_i,CP)$	Yes	Interaction between 3 birds	6.73
	20			Yes	Interaction between 5 birds	17.97
	50			Yes	Interaction between 38 birds	4.02
Traffic junction	10	Δc	v_i	No	No traffic jam.	-
			p_i			
			(v_i,p_i)			
	50	Δc	v_i	No	Traffic jam present - no relevant results	3.28
			p_i	Yes	Interaction between impolite drivers and other cars	2.01
			(v_i,p_i)	Yes	Interaction between impolite drivers and other cars	3.32
	100	Δc	v_i	No	Traffic jam present - no relevant results	3.31
			p_i	Yes	Interaction between impolite drivers and other cars	2.56
			(v_i,p_i)	Yes	Interaction between impolite drivers and other cars	4.78

Table 10: Summary of Results

life, flock of birds, and traffic junction models identifies interaction between specific model components in the model as the cause of emergence. For traffic junction models, reconstructability analysis is able to identify driver politeness as one of the direct causes of the emergence of traffic jam, to support both anecdotal evidence as well as previous studies using CCTV camera observations.

Our approach proposes a first step towards a formal method to determine the causes of emergence due to the model component interactions. Our case study shows the applicability of our approach but also highlights important avenues for future work. Firstly, the interaction information related to a particular emergent property can be saved into the repository and used it to augment live emergence analysis. This could help to identify emergence as it happens, if the model under study exhibits an interaction pattern similar to one encountered before. Next, techniques have to be developed to determine the attributes of interest for each model component, and the grouping attributes to allow for multi-dimensional analysis.

6. REFERENCES

[1] M. Bedau. Weak Emergence. *Philosophical Perspectives*, 11:375–399, 1997.

[2] R. Cavallo and G. Klir. Reconstructability Analysis of Multi-dimensional Relations: A Theoretical Basis for Computer-aided Determination of Acceptable System Models. *Int. Journal of General Systems*, 5:143–171, 1979.

[3] W. Chan, Y. S. Son, and C. M. Macal. Simulation of Emergent Behavior and Differences Between Agent-Based Simulation and Discrete-Event Simulation. In *Proceedings of the Winter Simulation Conference*, pages 135–150, 2010.

[4] W. K. V. Chan. Interaction Metric of Emergent Behaviors in Agent-based Simulation. In *Proceedings of the Winter Simulation Conference*, pages 357–368, Phoenix, USA, 2011.

[5] C. Chen, S. B. Nagl, and C. D. Clack. Specifying, Detecting and Analysing Emergent Behaviours in Multi-Level Agent-Based Simulations. In *Proceedings of the Summer Computer Simulation Conference*, 2007.

[6] L. Chi. Transplating Social Capital to the Online World: Insights from Two Experimental Studies. *Journal of Organizational Computing and Electronic Commerce*, 19:214–236, 2009.

[7] P. Cilliers. *Complexity & Postmodernism*. Routledge, 1998.

[8] P. Davis. New Paradigms and Challenges. In *Proceedings of the Winter Simulation Conference*, Orlando , USA, 2005.

[9] U. Fayyad and R. Uthurusamy. Evolving Data Into Mining Solutions for Insights. *Communications of the ACM*, 45, 2002.

[10] S. Floyd and V. Jacobson. The synchronization of Periodic Routing Messages. In *Proceedings of SIGCOMM*, pages 33–44, 1993.

[11] M. Gardner. *The Fantastic Combinations of John Conway's New Solitaire Games*. Mathematical Games, 1970.

[12] C. Gershenson and N. Fernandez. Complexity and Information: Measuring Emergence, Self-organization, and Homeostatis at Multiple Scales. *Complexity*, 2012.

[13] R. Gore and P. Reynolds. An Exploration-based Taxonomy for Emergent Behavior Analysis in Simulation. In *Proceedings of the Winter Simulation Conference*, pages 1232–1240, Miami, USA, 2007.

[14] R. Gore and P. Reynolds. Applying Causal Inference to Understand Emergent Behavior. In *Proceedings of the Winter Simulation Conference*, pages 712–721, Miami, USA, 2008.

[15] J. Holland. *Emergence, From Chaos to Order*. Basic Books, 1999.

[16] O. T. Holland. Taxonomy for the Modeling and Simulation of Emergent Behavior Systems. In *Proceedings of the 2007 Spring Simulation Multiconference*, pages 28–35, 2007.

[17] C. W. Johnson. What are Emergent Properties and How Do They Affect the Engineering of Complex Systems? *Reliability Engineering and System Safety*, 12:1475–1481, 2006.

[18] A. Kubik. Towards a Formalization of Emergence. *Journal of Artificial Life*, 9:41–65, 2003.

[19] E. Manley and T. Cheng. Understanding Road Congestion as an Emergent Property of Traffic Networks. In *Proceedings of International Multiconference on Complexity, Informatics and Cybernetics*, pages 109–114, 2010.

[20] J. C. Mogul. Emergent (mis)behavior vs. Complex Software Systems. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems*, pages 293–304, New York, USA, 2006.

[21] Occam. Occam reconstructability analysis: http://dmm.sysc.pdx.edu/, Last retrieved Feb. 2013.

[22] E. Page and J. Opper. Observations on the Complexity of Composable Simulations. In *Proceedings of the Winter Simulation Conference*, volume 1, pages 553–560, Phoenix, USA, 1999.

[23] M. Prokopenko, F. Boschetti, and A. J. Ryan. An Information-theoretic Primer of Complexity, Self-organization and Emergence. *Complexity*, 15:11–28, 2009.

[24] K. K. Ramakrishnan and H. Yang. The Ethernet Capture Effect: Analysis and Solution. In *Proceedings of the IEEE Local Computer Networks Conference*, Minneapolis, USA, 1994.

[25] C. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Proceedings of ACM SIGGRAPH*, pages 25–34, 1987.

[26] A. K. Seth. Measuring Emergence via Nonlinear Granger Causality. In *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 545–553, 2008.

[27] C. Szabo, Y. Teo, and S. See. An Integrated Approach for the Validation of Emergence in Component-based Simulation Models. In *Proceedings of the Winter Simulation Conference*, pages 2412–2423, 2012.

[28] C. Szabo and Y. M. Teo. Semantic Validation of Emergent Properties in Component-based Simulation Models. *Ontology, Epistemology, and Teleology of Modeling and Simulation Ð Philosophical Foundations for Intelligent MS Applications*, pages 319–333, 2012.

[29] Y. Teo and C. Szabo. CODES: An Integrated Approach to Composable Modeling and Simulation. In *Proceedings of the 41st Annual Simulation Symposium*, pages 103–110, Ottawa, Canada, 2008.

[30] M. Treiber and A. Kesting. *Traffic Flow Dynamics*. Springer, 2013.

[31] B.-Y. Yaneer. A Mathematical Theory of Strong Emergence using Multiscale Variety. *Complexity*, 9:15–24, 2004.

[32] M. Zwick. An Overview of Reconstructability Analysis. *Kybernetes*, 33:877–905, 2004.