

# Semantic Validation of Emergent Properties in Component-Based Simulation Models

Claudia Szabo<sup>1</sup> and Yong Meng Teo<sup>2</sup>

<sup>1</sup> The University of Adelaide,  
Adelaide, South Australia, Australia

<sup>2</sup> National University of Singapore,  
Singapore, Singapore

**Abstract.** Advances in composable modeling and simulation have facilitated the development and our understanding of more complex models. As a result, the representation, identification and validation of emergence is becoming of increasing importance because emergent properties can have a negative effect on the overall system behavior. Despite a plethora of definitions and methods, a practical approach to identify and validate emergent properties in newly composed simulation models remains a challenge. This chapter reviews current approaches and presents a new approach for identifying emergent properties in component-based systems. Using a simple example of a flock of birds model, we compare and contrast three main approaches: *grammar-based*, *variable-based* and *event-based*. Lastly, building on our previous work on formal semantic validation of model behavior, we present a new *objective-based approach* for semantic validation of emergent properties in composable simulation.

## 1 Introduction

*“The whole is greater than the sum of its parts” - Aristotle*

Complex systems often exhibit properties that are not easily predictable by analyzing the behavior of their individual, interacting components [13, 15]. These properties, called *emergent properties*, are increasingly becoming important as software systems grow in complexity, coupling, and geographic distribution [2, 12, 13, 15]. Examples of emergent properties include connection patterns in social network data analysis [7], trends in big data analytics [8], and power supply variation in smart grids due to provider competition [4]. More malign examples of emergent properties in computer systems are Ethernet capture effect [17], router synchronization

problems [9], and load-balancer failures in a multi-tiered distributed system [15] among others.

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 pose an interesting challenge. Despite ongoing research interest since the 1970s, most approaches focus mainly on the post-mortem observation of emergence in various biological, social, and AI systems, and less on measuring and advancing our understanding in the cause-and-effect of emergence. A plethora of examples of emergent properties have been identified and classified but few instances have been measured and explained [6, 12, 14, 15].

In this chapter, we present a new approach to identify and validate emergent properties as part of semantic composability validation. In validation, it is important to distinguish between expected behavior that stems from the interactions of the underlying components of a model, and emergence behavior or unexpected behavior arising from seemingly unrelated phenomena. While simulation validation demonstrates that a simulation meets expected behavior, emergent properties validation focuses on showing that the unexpected behavior is valid (or invalid) for a given set of conditions. In section 2, we review three key approaches to identify emergence: *grammar-based*, *variable-based*, and *event-based*. We discuss their advantages and limitations and show how these could be used in a simple example of a bird flocking model. Section 3 presents an objective-based validation approach for the semantic validation of emergence. In this approach, a meta-component describes each sub-component of the system. The meta-component includes among others a specification of the objective that the sub-component achieves. Using our approach, we next compute the entire system state and compare it with an objective-based reconstruction of the system state in the absence of interactions between sub-components. Section 4 summarizes this article.

## 2 Emergent Properties and Examples

An emergent property can be defined as “a property of an assemblage that could not be predicted by examining the components individually” [2]. Common characteristics of emergence include: radical novelty (features not previously observed in systems); coherence or correlation (meaning integrated wholes that maintain themselves over some period of time); a global or macro “level” (i.e. there is some property of “wholeness”); it is the product of a dynamic process (it evolves); and it is “ostensive” (it can be perceived). The fundamentals behind understanding different types of emergence lie in the assumption that in any component-based system there is a *micro-level*, the abstraction level of each individual component, and a *macro-level*, the abstraction of the composed model as a whole. Micro-level properties are usually measured by observing the component states such as the collection of all

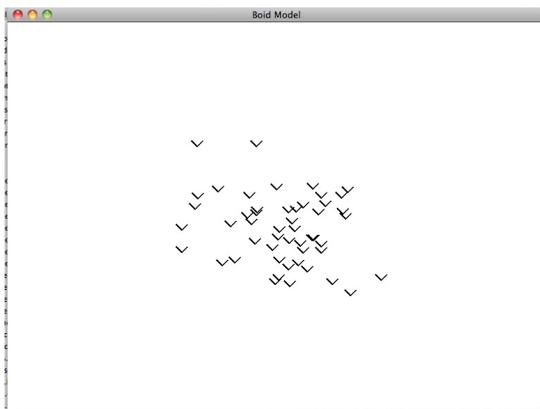
variables and their values, of each system component. In contrast, the macro-level properties can be measured either as an aggregation of all the states of the system sub-components, or by observing the overall system behavior and trends such as the change from non-flocking to flocking behavior in birds.

Three main types of emergence have been identified, namely *nominal*, *strong*, and *weak* [2]. In *nominal emergence*, the macro-level depends on the micro-level in the straightforward sense that the whole is dependent on their constituents. *Strong emergence* is a more powerful definition that assumes nominal emergence, but introduces *downward causation*, which can be informally defined as the influence of the macro-level on the micro-level. In contrast, *weak emergence* states that given the properties of the parts and the interaction rules among them, it is not *trivial* to infer the properties of the whole. In this context, trivial is taken to mean “by-hand” human calculations, and in order to identify weak emergence, one needs a computer model and its simulation.

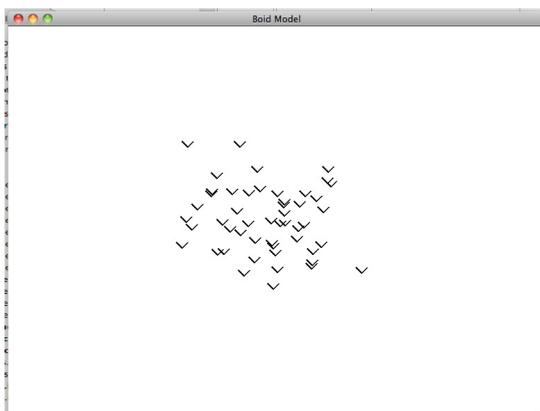
In the following, we focus on the main techniques and procedures to identify weak emergence in component-based simulation models. In component-based complex systems, emergence validation approaches are classified in three key categories, namely, *grammar-based*, *variable-based* and *event-based*. Most approaches, such as variable-based and event-based methods, assume that there exists an observation of emergence or irregularity prior to the validation exercise, and aim to identify the cause of emergence. The grammar-based approaches aim to identify emergence on the fly, by computing the difference between a system state obtained by the composition of sub-systems *with* and *without* interactions respectively. This method does not require a-priori observation of the system to identify possible emergent properties or behaviors, which makes it suitable for large systems where such observations are almost impossible. However, the nature of the formalism and the computation of the system states make it difficult to scale, as we will see below.

## 2.1 Approaches

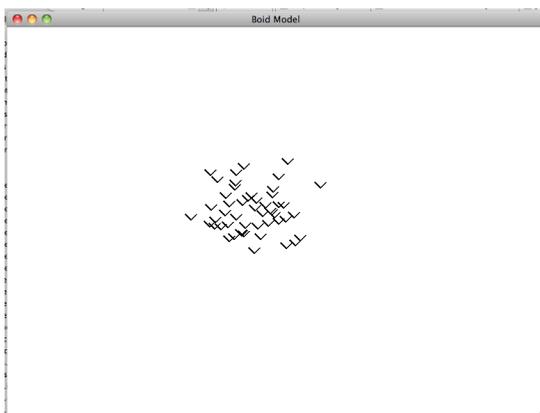
In this section, we discuss three main types of emergence validation approaches, namely, *grammar-based*, *variable-based* and *event-based*. We present a theoretical overview of each approach and discuss how it can be applied to a simple model of a flock of birds, also known in the literature as the boid model [18]. Each component abstracts a moving bird, which changes its position based on a set of simple rules that defines its current position and the position of the other birds in the flock. These rules are (i) *separation* - individual bird steer to avoid crowding the other birds in the flock (ii) *alignment*- individual bird steer towards the average herding of local flockmates and (iii) *cohesion* - individual bird moves towards the average position of local flockmates. The boid model has been shown to exhibit emergent behavior of flocking, and flocking after encountering an obstacle, when the flock splits and reunites. Fig. 1 shows a screenshot of flocking in our implementation.



(a) Initial Scattered State



(b) Initial Flocking



(c) Flocking

**Fig. 1** Visualization of Flocking in a Boid Model

### 2.1.1 Grammar-Based Approach

In grammar-based methods, two grammars,  $L_{WHOLE}$  describes the properties of the system as a whole, and  $L_{PARTS}$  describes the properties obtained from the reunion of the parts [14]. Kubik [14] proposes the use of grammar systems, which are symbolic devices composed from a set of grammars that interact with each other through tapes on which each grammar writes symbols. A formal *grammar* is a set of rules that governs the formation of *words* using a set of *symbols*. This paradigm applies easily to multi-agent systems, where each agent can be represented by a grammar and the behavior of an agent is represented by how it changes the symbols on the common tape.

Emergence is defined as the difference between the properties of the system as a whole  $L_{WHOLE}$ , and the reunion of the properties of the system parts,  $L_{PARTS}$  [14]. To derive  $L_{PARTS}$ , Kubik [14] proposes the superimposition of each agent language defined by the grammar system. Informally,  $L_{PARTS}$  is defined by the sum of the changes or conditions the agents bring about the environment if they would act individually in the system. This is obtained by using a superimposition of all the words that the agent grammars produce. In this superimposition,  $L_{PARTS}$  is formed using a reunion operator for all possible permutations of words created following rules that give higher priority to the symbols generated by the agent grammar, and less priority to the system symbols. While  $L_{PARTS}$  uses the superimposition operator to highlight the behavior of agents without considering the agent interaction with the environment,  $L_{WHOLE}$  is obtained by taking all the symbols written by all agents on the tape.

More formally, consider two words  $W_1 = a_1a_2\dots a_n$  and  $W_2 = b_1b_2\dots b_m$ , and their superimposition  $W_{superimp} = c_1c_2\dots c_l$ . We have the following:

1. if  $n \leq m \Rightarrow l = n$  else  $l = m$
2. if  $a_i \in V_A \Rightarrow c_k = a_i$
3. if  $a_i \in V_E$  and  $b_j \in V_E \Rightarrow c_k = a_i$
4. if  $a_i \in V_E$  and  $b_j \in V_A \Rightarrow c_k = b_j$
5.  $a_i = \varepsilon \Rightarrow c_k = b_j$
6.  $b_j = \varepsilon \Rightarrow c_k = a_i$

Thus,  $L_{PARTS} = \{W_1superimpose(W_2superimpose(W_3superimpose\dots W_n)\cup \dots W_nsuperimpose(W_{n-1}superimpose\dots(W_2superimposeW_1))\}$ . This approach is exemplified using very simple examples of a four-by-four Game of Life glider pattern [10]. However, the calculation of  $L_{PARTS}$ , which requires computing the reunion of all permutations of superimposing all of the words produced by agents is computationally expensive and might not scale well.

In the following, we discuss the application of this approach to identify emergent properties in the boid model. The pseudo-code for the grammar-based approach is presented in Fig. 2. As shown, the main difficulty in applying this pseudocode to a boid model lies in identifying the languages generated by each agent in the system,  $L(A_i)$ . We divide the drawing panel shown in Fig. 1 into a grid that is small enough such that at any point in time, each grid cell is only occupied by a single bird. For



As discussed above,  $L(BM)$  can be viewed as the language resulting from the interaction of all agents in the system, i.e.  $L_{WHOLE}$ .

To determine  $L_{PARTS}$ , we employ the superimposition operator in Section 2.1.1. Specifically, consider two agents,  $A_1$  and  $A_2$  that generate the following languages

$$L(A_1) = \left\{ \begin{array}{cccc} e e e e & e e e e & e e e e & e e e e \\ e B e e & e e B e & e e e B & B e e e \\ e e e e' & e e e e' & e e e e' & e e e e \\ e e e e & e e e e & e e e e & e e e e \end{array} \right\} \text{ and}$$

$$L(A_2) = \left\{ \begin{array}{cccc} e e e e & e e e e & e e e e & e e e e \\ e e e e & e e e e & e e e e & e e e e \\ B e e e' & e B e e' & e e B e' & e e e B \\ e e e e & e e e e & e e e e & e e e e \end{array} \right\}$$

with and without interaction respectively. The result of the superimposition of these two languages is

$$L_{sum} = L(A_1) \text{ superimpose } L(A_2) \cup L(A_2) \text{ superimpose } L(A_1) \Rightarrow$$

$$L_{sum} = \left\{ \begin{array}{cccc} e e e e & e e e e & e e e e & e e e e \\ e B e e & e e B e & e e e B & B e e e \\ B e e e' & e B e e' & e e B e' & e e e B \\ e e e e & e e e e & e e e e & e e e e \end{array} \right\}$$

It is important to highlight here that in this case it is evident that the superimposition does not contain interaction. For example, if interaction was considered, the third word in the sequence would contain  $B$  symbols on the same line.

The language generated while considering the interaction among agents is richer than individual agents without interactions.  $L_{WHOLE}$  is richer both in terms of the number of words and in the density of non-environment symbols, than  $L_{PARTS}$ . As such,

$$L(BM) - L_{sum} \neq \emptyset$$

The definition and set difference between  $L(BM)$  and  $L_{sum}$  provides a straightforward formal method for defining and identifying emergence. However, it is difficult if not impossible to pinpoint which of the elements in the  $\{L(BM) - L_{sum}\}$  set represents emergence, and what was its exact cause. Moreover, computing  $L_{sum}$  is computationally expensive as the superimposition operator has to consider all the combinations of words generated by all the agents in the system.

## 2.1.2 Variable-Based Approach

In variable-based methods, a specific variable is chosen to describe emergence. Changes in the values of this variable are said to signify the presence of emergent properties [18]. For example, changes in the centre of mass of a bird flock could be used as an example of emergence in bird flocking behavior, as shown in

[18]. 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.

According to the definition of *Granger causality*, a variable  $Y$  causes a variable  $X$  if the inclusion of past observations of  $Y$  reduces the prediction error of  $X$  in a linear regression model of  $X$  and  $Y$ , as compared to a model that only includes  $X$ . Another important definition is that of *G-autonomy*, in which the focus is on whether past observations of a variable  $X$  influence the current observation of  $X$  more than the values of other variables  $Y$  in the system. *G-emergence* is defined based on Granger causality and G-autonomy.

A macro-variable  $M$  is *G-emergent* from a set of micro-variables  $m$  iff (i)  $M$  is G-autonomous with respect to variables  $m$  and (ii)  $M$  is Granger causal with respect to  $m$ . In other words, a macro-variable could be G-emergent from a set of micro-variables if there are hidden or latent influences that are not evident in linear regression.

This approach provides a clear and easily measurable process to identify emergence because it looks at measurable quantities found in the system state, which is defined as the reunion of all sub-systems states. However, finding a good variable to describe a system can be a difficult task that requires system expert intervention. Moreover, Granger causality is designed to handle only pairs of variables, and might not apply when the macro-variable depends on more than one micro-variable.

We apply this approach to the flock of birds model shown in Fig. 1. The variable-based approach uses a Matlab software package [1] to calculate G-autonomy ( $ga_{M|m}$ ) and G-causality ( $gc_{M|m}$ ), between a macro variable ( $M$ ) and a set of micro variables ( $m$ ). The measure of emergence is then calculated as

$$ge_{M|m} = ga_{M|m} \left( \frac{1}{N} \sum_{i=1}^N gc_{m_i \rightarrow M} \right)$$

Towards measuring the G-emergence of the center of mass of the flock of birds, we use the pseudo-code presented in Fig. 3, which follows closely the description in [18].

Our aim is to establish whether the coordinates of the center of mass of the flock of birds  $CM(CM_x, CM_y)$  are G-emergent from the coordinates of each individual in the flock  $(x_i, y_i)$ , following a number of observations (*obs*) of these coordinates. Towards this, we construct *mat\_x* and *mat\_y*, a matrix with the coordinates of all individuals in the flock, as well as the center of mass, on the x-axis and y-axis respectively. The rows of the matrices represent the variables, and the columns the observations. The data is then pre-processed to reflect the distance from the  $(x, y)$  coordinates of the individual to the center of the environment, in *mat\_dist*. The pseudo-code returns a matrix in which values closer to one represent high G-emergence, as discussed in [18].

1. Collect `mat_x`
2. Collect `mat_y`
3. Calculate `mat_dist`
4. For each variable  $i$ ,
  - calculate  $gc_{m_i \rightarrow M}$ : `[gci] = cca_granger_regress(mat_dist, obs)`
5. Calculate  $ga_{M|m}$ : `[ga] = cca_autonomy_regress(mat_dist, obs)`
6. Calculate  $ge = 1/N * \text{dot}(gc1+gc2+\dots+gc_N)$

**Fig. 3** Pseudo-code for the Calculation of G-emergence

### 2.1.3 Event-Based Approach

In event-based methods, behavior is defined as a series of events which change a system or sub-system states [5]. The motivating example behind this work is that often when a macro-level property is constructed from the aggregation of sub-system states, there is a loss of information with respect to the *cause* of the emergent behavior. In particular, it is not possible to establish which sub-system interaction is responsible for the current behavior.

Towards this, the authors propose the definition of simple and complex event types. A *simple event* type signifies a change in a sub-system state. It is associated with a transition and has a duration. A *complex event* is defined as being either a simple event or comprises two complex events linked by a relationship. This relationship is a temporal operator (meaning that there is a temporal relationship between the two complex events) that can optionally have descriptions of constraints related to the environment or to the state of the two sub-systems.

Based on the above definitions, emergent behavior is defined beforehand as a sequence of event types, as shown in Fig. 4. A simulation is run and the appearance of the sequence defining emergent behavior is verified. This is formally done by representing the complex event types as a directed multi-graph, where the nodes represent various event instances in the complex event type and the directed arcs denote the relationships between two events. The simulation is also represented as a directed graph  $S_1$ . A complex event type is said to appear in the simulation if a sub-graph of the simulation graph can be proved to be isomorphic with the complex type graph. This provides an overview of the sequence of interactions that led to the appearance of an event or property.

For the flock of birds model, a complex event that represents emergence is *bird\_flocking*, represented as a temporal sequence of simple events *bird\_move<sub>i</sub>*, where  $i = 1 \dots n$  represents the id of the birds in the flock.

$$bird\_flocking = bird\_move_1 \rightarrow bird\_move_2 \rightarrow \dots \rightarrow bird\_move_n$$

Running the simulation of the flock of birds clearly identifies this sequence of events. However, a key assumption is that the description of emergence exists

1. Describe components as a collection of simple events  $s_e$
2. Identify complex event  $C_e = s_{ei} \otimes s_{ej}$
3. Run the simulation of the complex system and obtain the sequence of transitions from system states as  $S_1$
4. Represent the system state multi-graph from  $S_1$  and identify emergence

**Fig. 4** Pseudo-code for Identifying Complex Event Types

beforehand, which is not always the case in real-life scenarios, where emergence is something not seen or predicted before.

## 2.2 Discussion

The application of emergence validation approaches to the simple flock of birds model highlights a few important issues. The approaches described above can be categorized into *a-priori* and *a-posteriori* methods. In *a-priori* methods such as the variable-based and event-based approaches, there is a need to identify a variable or a complex event that defines the emergent property. The identification of this variable is manual and might not be straightforward for more complex examples. In *a-posteriori* methods such as the grammar-based approaches, a formalism guides the identification of emergence as a difference between the outcome of the interaction among the components in the system, and the outcome calculated if no interaction among components occurs. Ideally, the latter approach would be suitable for large complex systems where a single variable to define emergence is difficult to find. However, the limiting factor in these approaches is the formalism itself, which needs a high level of abstraction, as shown in the example.

It is crucial for the validation of emergence to have a micro-macro separation between the abstractions employed in the system model. The identification of micro and macro variables to describe a system is in most cases difficult to automate. Thus, a variable-based approach is suitable when there is a single variable that can characterize an emergent property, namely, the center of mass in a flock of birds. In this case, it is relatively straightforward to mathematically calculate the causality between this variable and the other parameters of individuals in the system. However, an important assumption is that the variable is identified beforehand to characterize emergence. Moreover, several simplifications need to be established on a case-by-case basis, and as such might make this approach difficult to automate.

Grammar-based approaches seem to solve this issue. However, to the best of our knowledge, current studies only look at applying this approach to the “Game of Life” model, in which a binary state (dead or alive) characterizes each agent. As we have seen in applying this method to the flock of birds example, several limiting abstractions have to be in place, because agents have a more complex state characterized by speed, heading, and position among others. These abstractions might result in a loss of precision in identifying emergence. In addition, the

computational complexity of calculating the languages generated by the agents, with and without interaction, seems to be exponential in the number of words generated by each agent. To the best of our knowledge, there is currently no study to analyze the computational complexity of this method.

In the following, we propose an a-posteriori, variable-based method that aims to address these issues. In our objective-based approach, we propose to define each system component in terms of the objective it aims to achieve. These objectives are defined as the outcome of a finite-state machine that models each component and are used in the system simulation. Next, we simulate the complex system and, at each simulation step, analyze the system state. We compare the simulated system state with a calculated system state using reconstructability analysis. In reconstructability analysis, component variables and objectives are used to calculate a system state without considering the interactions among the components, and the components with the environment.

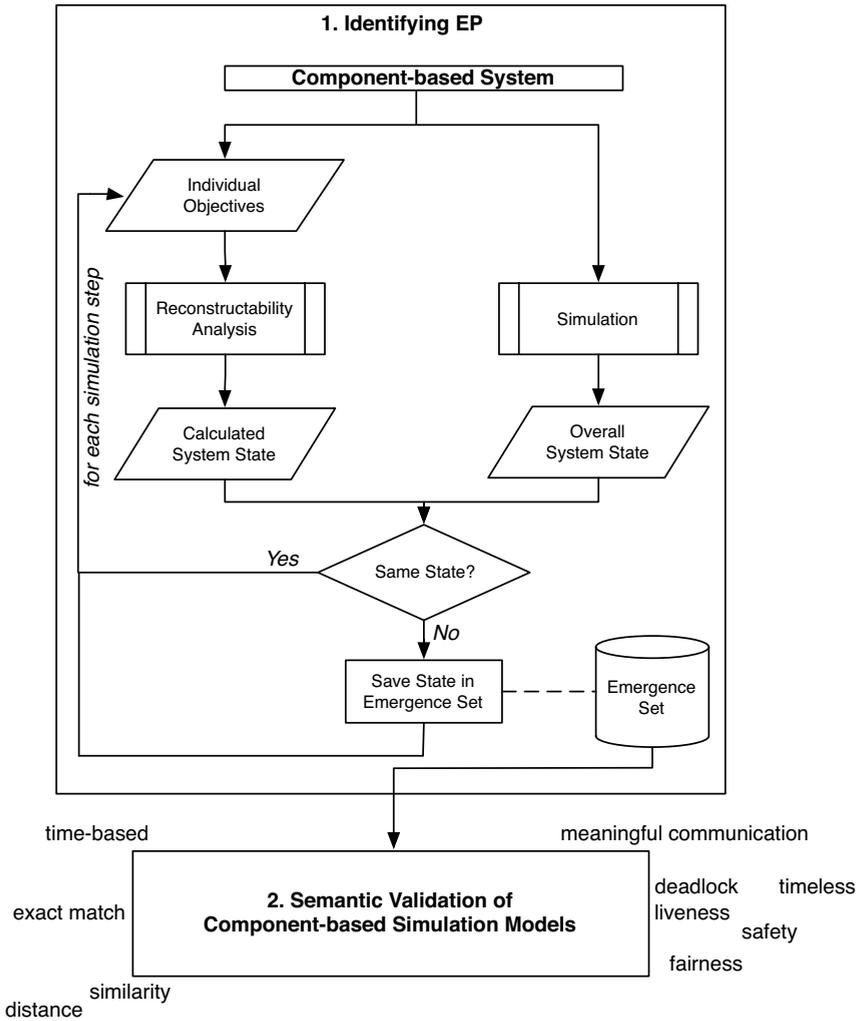
### 3 Emergence in Component-Based Model Development

In the modeling and simulation community, the possibility of emergent behavior in component-based model development has been highlighted since the 1990's by Page and Oppen [16]. They propose a slightly different definition of emergent behavior from systems theory. Let  $a$  and  $b$  denote two given components and its 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*. From a more practical perspective, Gore and Reynolds propose to identify the exact cause, with respect to variables and specific position in the source code, of parameter values that have not been encountered before [11]. Programming language techniques such as static analysis are employed to determine the variable(s) in the source code that result in the new value. However, this approach requires an expert to identify and detect a new or un-encountered variable value, as well as direct access to the source code for analysis.

#### 3.1 Objective-Based Approach for Identifying Weak Emergence

We propose to focus on the representation of systems in terms of objectives and properties, to facilitate the automated validation and identification of emergence. The focus of our objective-based approach is two-fold: (a) firstly, a composition is defined as a “sum” of its constituents; and (b) we define objectives to identify emergence. We then propose a semantic validation method to validate the component-based system.

Our objective-based validation approach relies on a definition of a system component that focuses on *what* rather than *how*. Each system sub-component is defined using the objectives it achieves as shown in Fig. 5. For example, a component property for a bird model would be “Fly north-bound with an average speed of 15 km/hour”. These objectives, defined as the outcome of a finite-state machine (FSM),



**Fig. 5** Objective-based Approach for Identification and Validation of Emergent Properties

defines each component and is used in the system simulation. Each transition in the FSM has a post condition which specifies the objective, in terms of variable values, that should be achieved by the transition. Next, at each time-step during the simulation run, we reconstruct the composed complex system from its sub-systems. This is done using a sub-problem of reconstructability analysis [3], which looks at reconstructing a complex system from variables defining its sub-systems. Next, we compare this theoretical, *calculated state* with the *simulation state*. If there is an unacceptable deviation in the observed parameters, we highlight this state as a

possible emergence state and add it to an *emergent set*. We repeat this for the entire simulation. At the end of the process, the emergent set is shown to the user.

The component-based model is then semantically validated. We propose to validate the component-based system model using our deny-validity approach [20, 21]. Towards the semantic validation of composed simulation models, our deny-validity approach subjects the composition to a battery of tests that either discard a composed model as invalid, or increase the credibility of the model that is not eliminated [19]. We first eliminate models that have invalid model properties through a feasible process that uses support from model checking and ontologies. At this stage, the validation process focuses on discarding composed models in a three-step approach. Firstly, the component interoperability with respect to exchanged data is validated, using semantically-sugared attribute values from our proposed component-based ontology. Secondly, we employ model checking to validate all possible interleaved execution schedules. From a practical perspective, we consider timeless transitions. Thirdly, we introduce time and validate a meta-simulation of the composed model, using properties specified by the model composer. However, models that pass the first validation stage may have valid properties but may still be invalid when compared with a reference model. In the next stage, our novel time-based formalism for the representation of the composed model supports the semantic comparison between a composed model and a reference model [21]. A model component is represented as a mathematical function of time and states. We introduce formal definitions of validity that consider closeness with respect to a reference model. We propose a semantic metric relation to evaluate this closeness, considering attribute and state relations in our proposed component-based ontology. The time-based formalism permits the validation of composed models with complex structures, but at increased computational cost. In this process, we incorporate and highlight the states from the emergent set to understand the sequence of events leading to them.

## 4 Concluding Remarks

This chapter presents state-of-the-art approaches for the identification and validation of emergent properties. The approaches are classified in four categories, namely, *grammar-based*, *variable-based*, *event-based*, and our proposed *objective-based*. Grammar-based approach attempts to formalize emergence as the difference between the language generated by the interaction of the individual components and with its environment, and the language generated by each individual in part. Variable-based methods define a system-wide variable as emergence to determine the causality relation between that variable and individual parameters. In event-based approach, a system-wide complex event is defined to determine the sequence of individual simple events that generate it. Lastly, objective-based methods use simulation and reconstructability analysis to identify differences between the desired system state and the actual system state.

Using a simple example of a flock of birds model, we studied these approaches. We showed that the variable-based approach is suitable when a single variable

defines an emergent property such as the center of mass in a flock of birds. In this case, it is relatively straightforward to mathematically calculate the causality between this variable and the other parameters of individuals in the system. However, an important assumption is that the variable is identified *a-priori* to characterize emergence. In addition, several simplifications have to be established on a case-by-case basis, and thus render this approach difficult to automate. In contrast, the grammar-based approach does not need an *a-priori* identification of an emergent variable. However, to the best of our knowledge, current studies apply this approach only to the “Game of Life” system, using a binary state (dead or alive) to characterize each agent, and thus the system can be easily abstracted on a two-dimensional grid, inline with the nature of a grammar. This is not the case in a more complex model, such as the flock of birds, in which agents are characterized by more than one state such as speed, heading, and position among others. As demonstrated in our example, the complexity of the model results in an additional layer of abstraction, and this abstraction might result in a loss of precision in identifying emergence. Moreover, the computational complexity of calculating the languages generated by the agents, with and without interaction, seems to be exponential in the number of words generated by each agent. To the best of our knowledge, there is currently no study to analyze the computational complexity of this method. Lastly, the event-based approach requires an *a-priori* identification of emergence, defined at the macro-level as a sequence of events from the micro-level.

In conclusion, there has been active research in defining emergence and identifying various real-life examples of emergence, but formally identifying and validating emergent properties, and automating this process remains a key challenge. Current state-of-the-art approaches are limited in most cases to simple systems and have yet to demonstrate their usefulness in more complex systems of practical use.

**Acknowledgements.** This work is supported by the Singapore Ministry of Education under the grant T1251RES1114.

## References

1. Kubik, A.: Causal Connectivity Analysis MATLAB toolbox, <http://ccatoolbox.pbworks.com/w/page/5805266/FrontPage> (last retrieved January 2012)
2. Bedau, M.: Weak Emergence. *Philosophical Perspectives* 11, 375–399 (1997)
3. Cavallo, R., Klir, G.: 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)
4. Chan, W., Son, Y.S., Macal, C.M.: Simulation of Emergent Behavior and Differences Between Agent-Based Simulation and Discrete-Event Simulation. In: *Proceedings of the Winter Simulation Conference*, pp. 135–150 (2010)
5. Chen, C., Nagl, S.B., Clack, C.D.: Specifying, Detecting and Analysing Emergent Behaviours in Multi-Level Agent-Based Simulations. In: *Proceedings of the Summer Computer Simulation Conference* (2007)

6. Chen, C.C., Nagl, S.B., Clack, C.D.: A Formalism for Multi-level Emergent Behaviours in Designed Component-based Systems and Agent-based Simulations. *Understanding Complex Systems* (2009)
7. Chi, L.: Translating Social Capital to the Online World: Insights from Two Experimental Studies. *Journal of Organizational Computing and Electronic Commerce* 19, 214–236 (2009)
8. Fayyad, U., Uthurusamy, R.: Evolving Data Into Mining Solutions for Insights. *Communications of the ACM* 45 (2002)
9. Floyd, S., Jacobson, V.: The synchronization of Periodic Routing Messages. In: *Proceedings of SIGCOMM*, pp. 33–44 (1993)
10. Gardner, M.: *Mathematical Games* (1970)
11. Gore, R., Reynolds, P.: Applying Causal Inference to Understand Emergent Behavior. In: *Proceedings of the Winter Simulation Conference*, Miami, USA, pp. 712–721 (2008)
12. Holland, J.: *Emergence, From Chaos to Order*. Basic Books (1999)
13. Johnson, C.W.: What are Emergent Properties and How Do They Affect the Engineering of Complex Systems? *Reliability Engineering and System Safety* 12, 1475–1481 (2006)
14. Kubik, A.: Towards a Formalization of Emergence. *Journal of Artificial Life* 9, 41–65 (2003)
15. Mogul, J.C.: Emergent (mis) behavior vs. Complex Software Systems. In: *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems*, New York, USA, pp. 293–304 (2006)
16. Page, E., Opper, J.: Observations on the Complexity of Composable Simulations. In: *Proceedings of the Winter Simulation Conference*, Phoenix, USA, vol. 1, pp. 553–560 (1999)
17. Ramakrishnan, K.K., Yang, H.: The Ethernet Capture Effect: Analysis and Solution. In: *Proceedings of the IEEE Local Computer Networks Conference*, Minneapolis, USA (1994)
18. Seth, A.K.: Measuring Emergence via Nonlinear Granger Causality. In: *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pp. 545–553 (2008)
19. Szabo, C., Teo, Y.: An Approach for Validation of Semantic Composability in Simulation Models. In: *Proceedings of the 23rd ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*, New York, USA, pp. 3–10 (2009)
20. Szabo, C., Teo, Y.: On Validation of Semantic Composability in Data-driven Simulation Models. In: *Proceedings of the 24th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*, Atlanta, USA, pp. 73–80 (2010)
21. Szabo, C., Teo, Y., See, S.: A Time-based Formalism for the Validation of Semantic Composability. In: *Proceedings of the Winter Simulation Conference*, Austin, USA, pp. 1411–1422 (2009)