Engineering Self-Organising Emergent Systems with Simulation-based Scientific Analysis

Tom De Wolf[1], Tom Holvoet[1], and Giovanni Samaey[2]

[1] AgentWise@DistriNet Research Group
[2] Scientific Computing Research Group
Department of Computer Science, KULeuven
Celestijnenlaan 200A, 3001 Leuven, Belgium
{Tom.DeWolf,Giovanni.Samaey,Tom.Holvoet}@cs.kuleuven.be

Abstract The goal of engineering self-organising emergent systems is to acquire a macroscopic system behaviour solely from autonomous local activity and interaction. Due to the non-deterministic nature of such systems, it is hard to guarantee that the required macroscopic behaviour is achieved and maintained. Before even considering a self-organising emergent system in an industrial context, e.g. for Automated Guided Vehicle (AGV) transportation systems, such guarantees are needed. An empirical analysis approach is proposed that combines realistic agent-based simulations with existing scientific numerical algorithms for analysing the macroscopic behaviour. The numerical algorithm itself obtains the analysis results on the fly by steering and accelerating the simulation process according to the algorithm's goal. The approach is feasible, compared to formal proofs, and leads to more reliable and valuable results, compared to mere observation of simulation results. Also, the approach allows to systematically analyse the macroscopic behaviour to acquire macroscopic guarantees and feedback that can be used by an engineering process to iteratively shape a self-organising emergent solution.

1 Introduction

In an industrial research project EMC^2 [1], we examine the possibilities of a *self-organising emergent* solution for an Automated Guided Vehicle (AGV) warehouse transportation system. In this case study a group of AGVs has to transport incoming loads from pick up locations to specific destinations in the warehouse. Experience has shown that the current centralised system has problems with scalability because it cannot handle many AGVs efficiently. Also, the system is not flexible, i.e. the current solution cannot handle frequent changes in the transportation problem and it needs to be customised and optimised each time the system is deployed. There is a need for a decentralised system that adapts itself to each different situation. In [2], both 'emergence' and 'self-organisation' are defined and a combination of both is a promising approach for a complex and dynamic system such as the AGV system. Other examples of systems where self-organised emergent solutions are advantageous are telecommunication networks, flexible manufacturing networks, and dynamic traffic networks.

The goal of engineering self-organising emergent systems is to acquire a system with a coherent macroscopic behaviour which meets the requirements and results solely from autonomous local activity and interaction. *Macroscopic* is defined as being observed as an overall or global pattern, structure, or behaviour of the system as a whole. Despite multiple initiatives, how to systematically build such a system remains an open issue. Today, such systems are mostly built in an ad-hoc manner. However, experience in the industrial project EMC^2 [1] revealed that before even considering a self-organising emergent system in an industrial context, guarantees are needed that the required coherent macroscopic behaviour is achieved and maintained.

This paper describes an approach that allows to systematically analyse the macroscopic behaviour to guarantee that the requirements are achieved. Realistic agent-based simulations are combined with existing scientific numerical analysis algorithms for dynamical systems. This combination leads to more reliable and valuable results, compared to mere observation of simulation results, because the analysis algorithm itself obtains the results on the fly by steering and accelerating the simulation process according to the algorithm's goal. In order to also achieve a more systematic approach for building self-organising emergent systems, it is proposed to integrate the analysis approach into the engineering process such that a constant feedback loop between scientific analysis and engineering shapes a self-organising emergent solution. In the end, a systematic simulation-based engineering process can be achieved.

The paper is structured as follows. In section 2 the AGV case study that is used throughout the paper is described and motivated. Then, section 3 describes the analysis approach and how to systematically apply it. Section 4 discusses how to systematically engineer a self-organising emergent system by exploiting scientifically founded analysis results. Finally, a conclusion and some directions for future work are given.

2 The Case Study: Automated Guided Vehicles

In an industrial research project EMC^2 [1], our group develops self-organising emergent solutions for AGV warehouse transportation systems. Experience revealed an industrial need for guarantees about the macroscopic behaviour. Therefore, the AGV case is used throughout the paper. The automated industrial transport system, that we consider, uses multiple transport vehicles. Such a vehicle is called an AGV and is guided by a computer system (on the AGV itself or elsewhere). The vehicles get their energy from a battery and they move packets (i.e. loads, materials, goods and/or products) in a warehouse. Each individual AGV is capable of only a limited number of local activities such as move, pick packet, and drop packet. The goal of the system is to transport the incoming packets to their destination in an optimal manner.

A simulator[3] is developed for such an AGV system that allows to execute the simulations needed in the analysis approach described later in this paper. The screen-shot in figure 1 illustrates the problem setting described above: the locations where packets must be picked up are located at the left of the figure, the destinations are located at the right, and the network in-between consists of stations and connecting segments on which the AGVs move. Segments are unidirectional. A bidirectional segment is constructed using two overlapping unidirectional segments. Packets are depicted as rectangular boxes and some of the AGVs shown hold a packet, others do not.

---

[3] http://www.cs.kuleuven.be/˜distrinet/taskforces/agentwise/agvsimulator/

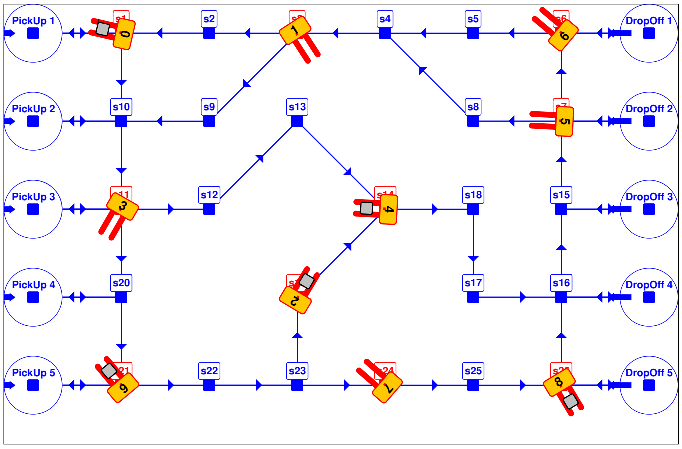


Figure 1. Screen-shot of AGV Simulator

The AGV problem is a dynamic problem with non-deterministic features (e.g. packets can arrive at any moment, AGVs can fail, obstacles can appear). The project with our industrial partner [1] has shown that a solution using a central server to control all AGVs cannot handle the frequent changes efficiently. The central server has to constantly monitor the warehouse and each AGV to detect and react to changes by steering each AGV. Because AGVs are moving constantly, reacting on changes has to occur instantaneously. The central server becomes a bottleneck in the presence of frequent changes. As a consequence, the current central solution is not scalable and can only handle a limited number of AGVs. Also, the system is not flexible when it has to be deployed, i.e. the system needs to be customised and optimised each time it is deployed in another warehouse. Therefore, larger and dynamic AGV systems require a self-organising emergent solution in which the AGVs adapt to the changing situations themselves by only using locally obtained information, local interactions, and local activity. However, before even considering a self-organising emergent solution for the AGV system, guarantees are needed that the required coherent macroscopic behaviour is achieved and maintained. Thus, an analysis approach integrated into the engineering process, that offers guarantees, is required.

3 The Analysis Approach

The main question here is how to know if a certain self-organising emergent system exhibits the required macroscopic behaviour. The analysis approach described in this section focusses on how to analyse the macroscopic behaviour of such a system and should result in guarantees about that macroscopic behaviour. Firm guarantees could be obtained if the system is modelled formally and the required macroscopic behaviour is proofed analytically. However, constructing a formal model and correctness proof of a complex interacting computing system is infeasible. Wegner [3] proves this based on the fact that computing systems using interaction are more powerful problem solving engines than mere algorithms. Interaction models are even so powerful that they can be denoted to be incomplete, in the mathematical sense. Completeness ensures that all possible behaviour is modelled. Wegner shows that one cannot model all possible behaviour of an interaction model and thus formally proving correctness of interactive models (e.g. self-organising emergent systems) is not merely difficult but impossible.

The alternative is to use an empirical and scientifically founded method to analyse the macroscopic behaviour, which is also advocated in [4,5]. Empirical analysis requires focussing on relevant properties and ignoring irrelevant ones. Macroscopic properties are typically quantified with measurable variables which we define as *macroscopic variables*. Using macroscopic variables, i.e. an incomplete representation of the system behaviour, is not problematic because physicists achieve their pragmatic goals of prediction and control also by dealing entirely with incomplete observable representations.

Section 3.1 focusses on what is important to analyse in self-organising emergent systems, i.e. which macroscopic variables. Before any analysis is done, the system needs to be modelled. Section 3.2 discusses the distinction between aggregate-based models and individual-based simulation models and why the latter is to be preferred. Section 3.3 describes the analysis approach that combines realistic individual-based simulation models with an existing arsenal of numerical analysis algorithms to analyse the behaviour of self-organising emergent systems. And finally, section 3.4 discusses interesting macroscopic properties in the AGV case.

3.1 Analysis of Self-Organising Emergent Systems: Trends

Before the analysis approach is outlined, we first need to define the results that are expected from the analysis of self-organising emergent solutions. What kind of macroscopic properties and variables are we interested in?

Self-organising emergent systems promise to be scalable, robust, stable, efficient, and to exhibit low-latency [6], but also behave non-deterministically. Even if the required macroscopic behaviour is achieved, the exact evolution is not predictable [6]. However, self-organising emergent systems exhibit *trends* that are predictable. A trend is defined to be the evolution of the macroscopic behaviour when the average is taken over a number of system runs. Due to the dynamics of self-organising emergent systems, robustness is preferred to an optimal macroscopic behaviour. Optimality is only achieved when the conditions in which the system operates remain rather static. But a static situation will never be reached in the presence of frequent changes. Preferring robustness above optimality implies that most temporal deviations from the required behaviour are allowed as long as the required behaviour is maintained in a trend, i.e. in the evolution of the average behaviour. Temporal deviations are often necessary to explore the space of possibilities, and to counteract the frequent system changes.

Therefore, the main results expected from an analysis of the macroscopic behaviour of self-organising emergent systems are statements about the macroscopic behaviour that ensure the required trend in the evolution of that behaviour. We define such statements as *macroscopic guarantees*. The analysis approach described in section 3.3 focusses on the analysis of trends in order to obtain macroscopic guarantees. Other issues of the dynamics of the system (i.e. microscopic behaviour, frequency of deviations and how large they are, etc) are also important but are outside the scope of this paper.

3.2 Modelling: Individual-based versus Aggregate-based

Before any analysis is done, a model of the analysis subject (i.e. the macroscopic behaviour) is needed. Generally, there are two kinds of models. First, there are so called *aggregate-based models* which are constructed using macroscopic variables as building blocks and defining the relationships between those variables. One implicitly assumes that the evolution of the macroscopic variables is the result of the behaviours of individuals in the system. Traditionally such models are *equation-based models* where the evolution of an equation represents the evolution of the macroscopic behaviour. Second, there are *individual-based models* which consist of a set of agents that explicitly encapsulate the behaviours of the various individuals that make up the system. The macroscopic behaviour is then observed by running a simulation of the model.

In the context of self-organising emergent systems, individual-based models have a number of advantages with respect to equation-based models (see [7]). The most relevant advantages are:

- Individual-based models are easier to construct. They are a natural 1-on-1 mapping between the system and the model. Often such a model just equals the system. Individual-based models are appropriate for domains characterised by a high degree of localisation and distribution which is the case with self-organising emergent systems. Equation-based models are often infeasible because such mathematical formal models are impossible to construct for complex interacting systems as discussed at the beginning of section 3 and in [3]. Even if they are possible, the model would be too complex for reasonable manipulation and comprehension.
- Individual-based models make it easy to adjust the model in order to play "what-if" games without translating into or constructing a new equation-based model.
- Equation-based models may yield less realistic results compared to an individual-based model. This is mainly due to the simplification that is often required to construct an equation while every detail in the individual behaviours can have a significant impact on the macroscopic result. For example, Wilson [8] offers a detailed study that compares individual-based models and equation-based models for a predator-prey system and finds that the equation-based models can result in qualitatively different behaviours compared to the real behaviour, especially due to the stochastic behaviour in the individual-based simulation.

On the other hand, equation-based models are very popular mainly due to the availability of a whole arsenal of numerical tools and techniques for analysing system dynamics. In contrast, individual-based models lack such scientific techniques. Mere observations of individual-based simulations only results in reliable guarantees if a lot of simulations are executed for a long time. In the context of self-organising emergent systems, simulations are expensive and thus the amount of simulation time needs to be minimised.

Parunak [7] argues that one should choose for either individual-based modelling or for equation-based modelling depending on the problem at hand. However, the analysis approach described in section 3.3 combines the advantages of both approaches, i.e. realistic individual-based simulation models and aggregate-based numerical analysis. The need for equations is eliminated and the simulation process is accelerated, i.e. according to the the goal of the analysis algorithm the amount of expensive simulation time is limited to only essentially necessary simulations.

3.3 “Equation-Free” Macroscopic Analysis

This paper uses a *“equation-free” macroscopic analysis* approach [9,10], that combines numerical analysis algorithms and realistic individual-based simulation models. Traditionally, numerical analysis is applied to equation-based models. The macroscopic behaviour is modelled by an equation (a macroscopic equation) and numerical algorithms are used to obtain quantitative statements about the macroscopic properties. However, as discussed earlier, in complex and dynamical systems, deriving a macroscopic equation is often not possible, unless the system is very simple. The “equation-free” approach resolves this issue by replacing the equation-based model by a realistic individual-based simulation model. This approach also analyses simulation measurements, but, in contrast to mere observation of simulation results, the numerical analysis algorithms acquire the results themselves by steering the simulation process towards the algorithm’s goal. For example, a numerical algorithm could have as its goal to find a steady state behaviour if present, i.e. the measured behaviour converges to a single value. Another goal could be to extrapolate the behaviour as far as possible in time. The advantage is that the results are calculated on the fly and only those simulations are executed that are actually needed to obtain a specific result. The latter reduces the computational effort drastically compared to mere simulations where one typically simulates for a huge number of time steps starting from time step 0. The results are of equal or even better scientific value as the equation-based analysis because the same scientific numerical algorithms are used and possible discrepancies between the model and the real system dynamics (e.g. [8]) are avoided by using a realistic individual-based model . Note that numerical algorithms assume a rather smooth behaviour (i.e. a rather continuous evolution over time). As described in section 3.1, for self-organising emergent systems, the main focus is on analysing trends which are expected to evolve gradually.

The approach. The “equation-free” macroscopic analysis approach was proposed in [9,10]. The observation is that most numerical techniques have no explicit need for the macroscopic equation; all they need is a routine that *evaluates* the equation for a given value of the macroscopic variables. Once the equation evaluations are replaced with a suitable *simulation*, all the numerical analysis algorithms can be readily applied.

To achieve this, the following procedure (illustrated in figure 2), called a *“macroscopic time-stepper”* [9], is performed. First the initial values (x_i) for all the macroscopic variables under study are supplied to the analysis algorithm. Then, given those values, the *initialisation operator* (`init`) initialises a number of simulations accordingly, i.e. such that a measurement of the macroscopic variables after initialisation equals the given initial values. Because there are multiple ways to initialise a simulation for the same value of a macroscopic variable, these “degrees of freedom” need to be initialised randomly in multiple initialisations. Then the *simulations* (`simulate`) are executed for a predetermined duration using the individual-based simulation code. At the end, one *measures* (`measure`) the averages over all simulations of the values of the macroscopic variables (x_{i+1}), which are given to the analysis algorithm as a result. As such, the equation and its evaluation are replaced by the simulation code. Then the analysis algorithm processes the new results to obtain the next initial values. The

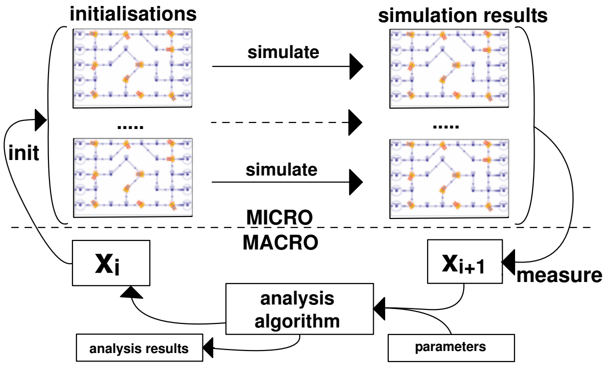


Figure 2. Equation-Free Accelerated Simulation guided by the analysis algorithm

algorithm itself decides on the configuration of the simulations such as the initial conditions and the duration. The `init-simulate-measure` cycle is repeated until the analysis algorithm reaches its goal.

There exists a whole arsenal of numerical algorithms to analyse system dynamics. One example is called the *projective integration algorithm* where the goal is to make a considerable acceleration of the simulations over time by minimising the number of needed simulation steps through extrapolation. Figure 3 illustrates the basic idea. First an initial value x_1 for the measured macroscopic variable is chosen by the analysis algorithm. Using the initial value x_1, a (set of) simulation(s) is initialised with microscopic value(s) X_1 (i.e. `init` operator on figure 3) and executed for a certain duration. At some points in time, one measures the new value for the macroscopic variable (i.e. `measure` operator on figure 3). The measure operator is repeated for a number of times such that enough successive values x_k are available for the projective integration algorithm to make the extrapolation step that skips m time steps of simulation. As such, a new value x_{k+1+m} is estimated with extrapolation, using a number of measured values (x_k and x_{k+1} in figure 3). Starting from the new value x_{k+1+m} the process is

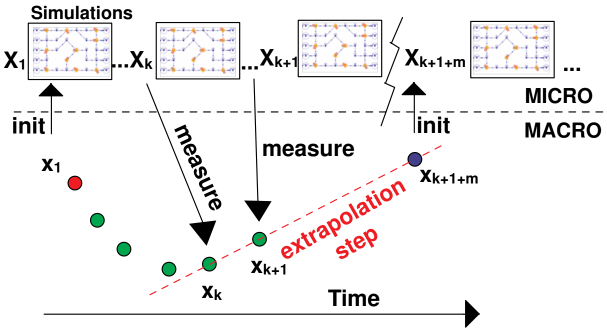


Figure 3. Equation-Free Accelerated Simulation over Time

repeated by initialising new simulations with microscopic value(s) X_{k+1+m}. As such an acceleration over time is achieved.

Simulations can also be accelerated in other ways. Suppose the goal is to obtain the steady state behaviour, i.e. we look for values of the macroscopic variables that remain constant as time evolves. Instead of computing the time evolution from time step 0 until the system stays at the steady state long enough, one uses numerical procedures that determine steady states in a more direct and efficient way, e.g. Newton's algorithm [11]. Denote the macroscopic time-stepper starting from an initial macroscopic value x_i, performing a simulation for a fixed duration, and measuring the new macroscopic value by $\Phi(x_i)$. Denote the measured value of the macroscopic variable after the simulation as $x_{i+1} = \Phi(x_i)$. The steady state x^* is then computed by solving the equation

$$\Phi(x^*) - x^* = 0 \qquad (1)$$

numerically by Newton's algorithm. This iterative method is illustrated in figure 4. The x-axis contains the initial values x_i and the y-axis the differences $|x_i - \Phi(x_i)|$ between the initial and the measured values for one cycle of the macroscopic time-stepper. First one chooses two initial values x_1 and x_2 close to each other. For each of them a `init-simulate-measure` cycle is done to get the measured values. Through these measured values an estimation for the derivative of the plot in figure 4 is used to extrapolate to a new estimation x_3 for the steady state. Then the cycle is repeated (x_3 - x_6) until the conditions and thresholds of the Newton algorithm decide to have reached a steady state x_7.

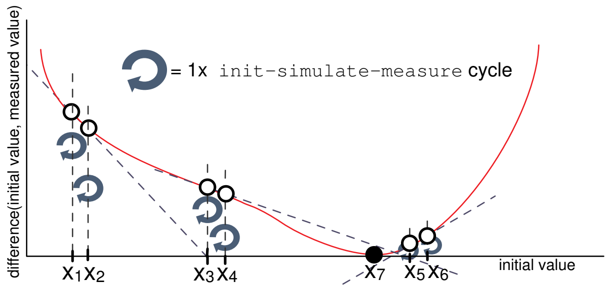


Figure 4. Equation-Free Newton Analysis

In contrast to mere observation of simulation results, only a limited set of rather short simulations are necessary to generate consecutive approximations for the steady state and a scientific algorithm objectively decides on the accuracy of the result. Note that the due to the equation-free approach equation (1) itself is not required. Newton's algorithm only needs the evaluation of the equation at certain points in time and this is replaced by a suitable simulation measurement. Following the same equation-free principle, e.g. in the presence of parameters, more general tasks, such as parameter optimisation or control can be performed as well [9]. As such, a more focussed and accelerated simulation-based analysis approach, guided by the analysis algorithm's goal, is used to obtain more reliable and valuable results than mere observation of simulation results. As such the proposed method constitutes a bridge between classical numerical analysis and microscopic (e.g. agent-based) simulation.

Road Map. There are a number of steps needed for the equation-free approach to work. An overview of is given with ant foraging based on pheromones as an example:

1. *Identification of macroscopic properties*. The goal of the analysis approach is to systematically acquire results that give macroscopic guarantees. A systematic approach often implies that one divides the problem into manageable subproblems. The macroscopic behaviour of a self-organising emergent system typically consists of a number of *macroscopic properties* that have to be maintained. For the ant foraging example, macroscopic properties such as the amount of food gathered, the directed movement of the ants, and the shape of the pheromone path are important. In a first step, each of the macroscopic properties are considered separately in the analysis approach and one has to identify which properties are important for the considered case study (examples for the AGV case are given in section 3.4).
2. *Identification of macroscopic variables*. In the context of self-organising emergent systems, the challenge is to find variables that are measures for the macroscopic properties under study. In other words, a *quantification of the macroscopic properties* in terms of measurable variables is needed. For example, using entropy to measure the concentration of ants on pheromone paths or to measure how focussed ants choose a direction is a possibility [12].
3. *Related macroscopic variables*. Are there other variables for macroscopic properties that influence the property under study? If so, then these variables have to be incorporated into the analysis process. Otherwise, the evolution of the system is not correctly and completely represented and analysed. An underlying assumption of the equation-free analysis approach is that a set of measurable variables are found that offer an adequate description of the macroscopic system dynamics. For example, only entropy of the concentration of ants is not enough. This variable omits the evolution of the pheromones which also influence the ant-movement. Therefore, variables are needed that capture the macroscopic evolution of the pheromones.
4. *Microscopic variables*. For each macroscopic variable, the corresponding variables of the simulation at the level of the individual entities in the system, that influence the macroscopic variable need to be identified. With ant foraging, this will include variables such as the exact positions of the ants, the strengths and positions of pheromones, and if an ant holds food or not.
5. *Measurement operator*. Define a measurement operator that measures the macroscopic variables from the microscopic variables in the simulation.
6. *Initialisation operator*. An operator needs to be defined that allows to initialise the microscopic variables of the simulation or multiple simulations to reflect the given values for the macroscopic variables. When there are degrees of freedom in the initialisation, these are initialised randomly and multiple simulations are considered to average this randomness. For example, re-initialising the ant system requires positioning each ant but given the concentration of ants, the exact positions allow some degrees of freedom.

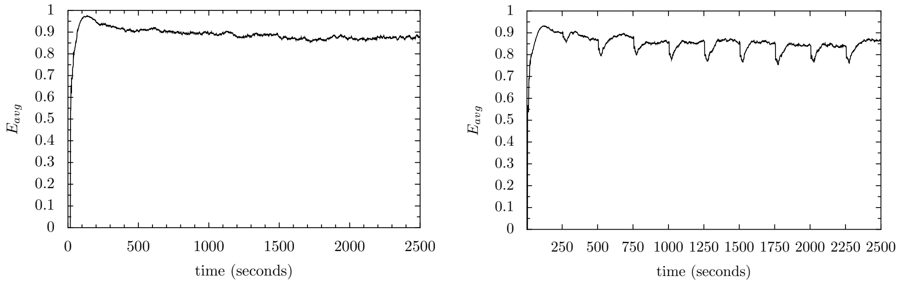
7. *Micro-macro scale separation*. For the equation-free approach to work and to be efficient, it should be checked that the microscopic variables evolve on a much faster timescale than the macroscopic variables that determine the macroscopic evolution. Thus, changes in the state of the individual entities in the system (e.g. movement of ants) need to be fast compared to the evolution of the overall system behaviour (e.g. changes in pheromone path shape or ant concentration). If this is not the case, then any error introduced by the extrapolation and/or initialisation procedure could significantly influence the results and hence create errors.
8. Define the different *steady scenarios* to analyse. A macroscopic guarantee that holds in all possible conditions in which a system can be executing is difficult, if not impossible to give. A *steady scenario* is defined as a setting for the system in which certain assumptions are made about the operational conditions (i.e. initial conditions, possible changes, and the frequency of change). This step of the road map identifies the system parameters that need to be modified in order to cover the range of possible operational conditions for the system. For example, one can consider steady scenarios where the system has a high utilisation load, a low utilisation load, or a scenario where there is a frequent oscillation between high and low utilisation loads. In the ant foraging example, parameters such as the number of ants involved, the evaporation rate of the pheromones, and the amount of food present can be modified. As a consequence, macroscopic guarantees are always given with respect to a specific steady scenario and for a specific macroscopic property (see step 1 of road map). A complete analysis result of the macroscopic behaviour then consists of multiple macroscopic guarantees.
9. *Analysis algorithm*. In the end, an analysis algorithm is to be chosen. Depending on the kind of data and the goals, e.g. finding the steady state behaviour, optimising the value of a system parameter (e.g. which evaporation rate is optimal for a certain scenario), controlling the operational modus, one selects a suitable numerical analysis algorithm (e.g. projective integration, Newton's method).

With self-organising emergent systems, an open issue is understanding how the macroscopic behaviour is accomplished by the individual entities. There is a gap between the result seen at the macroscopic level and the rules at the microscopic level that cause that result. Some of the steps in the road map are far from trivial, because some knowledge on how to bridge the micro-macro gap is required. Especially defining the measurement and initialisation operators is challenging. This requires to know how to represent a macroscopic property as a variable, how to measure it from the microscopic level and how to initialise a microscopic simulation to reflect a macroscopic variable. However, a successful application of the "equation-free" analysis road map can result in new insights in how the macroscopic level is related to the microscopic level. And, as shown in section 3.4, the approach offers the ability to check if a certain set of macroscopic variables completely captures the evolution of the macroscopic property.

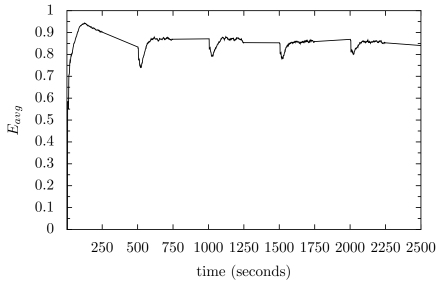
3.4 Macroscopic Variables in the AGV Case

In each application domain, the important macroscopic properties will be different. The requirements that have to be achieved by the macroscopic behaviour determine the important macroscopic properties and the desired guarantees about them. In the AGV case a number of issues are important, some examples:

- *Distribution of AGVs* over the factory floor is a macroscopic property from which one could require that on average the AGVs are equally distributed over the factory floor, i.e. a maximum coverage. In a recent paper [13] we used a spatial entropy measure as a macroscopic variable that reflects the distribution of AGVs.



(a) A normal begin-to-end simulation.
(b) A re-initialisation based simulation



(c) An extrapolated simulation

Figure 5. Different simulation processes for the Distribution of AGVs reflected by the average spatial entropy (E_{avg}) evolution (high entropy is equal distribution, low entropy is unequal distribution).

The results validated the described analysis approach. For example, in figure 5(a) the evolution of the average entropy during normal begin-to-end simulations is shown, i.e at each time step the average over 100 runs is taken. In figures 5(b) and 5(c) the same evolution is shown but obtained in two different ways. A first simulation process was executed in which simulations are done over 250 time steps, then the set of measured macroscopic variables (including entropy) is used to initialise a new set of simulations and again simulations of 250 time steps are done, and so on (figure 5(b)). The results show that a simulation based on re-initialisation with the chosen set of macroscopic variables reflects the correct average evolution compared to figure 5(a). Thus, the re-initialisation based simulation process allows to check if a chosen set of macroscopic variables is enough to reflect the evolution of the system, i.e. there are no other macroscopic variables needed to have a representative analysis (see step 3 of Road Map).

A second simulation process was executed in which the projective integration algorithm was used averaged over 100 simulation runs with a simulation duration and extrapolation step of 250 time steps. The results in figure 5(c) show that one can analyse the evolution of the average entropy with half the simulation time needed than normal begin-to-end simulation. As shown on figure 5, the distribution evolved to a steady state behaviour, i.e. a stable and almost equal distribution of the AGVs over the factory floor is achieved. More details can be found in [13]. Applying Newton's algorithm (see section 3.3) allows to confirm that this is a steady state behaviour scientifically (i.e. more accurate than mere observation) with only a few simulations.

- *Throughput* is the important characteristic in the AGV case. Throughput is defined as the number of packets transported in a certain time span (e.g. one hour). The challenge here is to identify macroscopic variables that allow to represent and measure the throughput evolution. Because throughput is a characteristic expressed over time and because the analysis approach expects the ability to measure the macroscopic variables at each time step, time-independent macroscopic variables are needed from which one can calculate the throughput as a post-analysis step.

  A possible set of macroscopic variables is the following:

  - The number of packets in transport (NBP_T), i.e. the number of AGVs currently holding a packet.
  - The number of packets in the queues at the pick-up locations (NBP_Q). Each pick-up location has a queue in which packets arrive. NBP_Q is the sum of all queue lengths.

  Assume that packets arrive in the pick-up location queues at a fixed arrival rate $AR_{\Delta t}$, i.e. the number of packets arriving in a time span of Δt. Then the throughput can be calculated as follows. Define the queue growth rate as the increase or decrease of total queue length in a time span of Δt, i.e. $QR_{\Delta t} = NBP_{Q,t} - NBP_{Q,t-1}$ with $NBP_{Q,t}$ the total queue length at time t. If the number of packets in transport is rather constant (i.e. $NBP_T = cte$) then throughput is defined as $T_{\Delta t} = AR_{\Delta t} - QR_{\Delta t}$. Another possible approach to calculate the throughput is to count the number of packets that are delivered at drop-off locations in a time span Δt. However, the number of drop-offs is a time dependent variable and re-initialising based on this variable at one moment in time is not possible and useful. The calculation given above only uses time independent variables, i.e. real state of the system.

It is clear that the biggest challenge in applying the analysis approach is to find suitable macroscopic variables that reflect the evolution of macroscopic properties for which one needs certain guarantees. Based on macroscopic variables a re-initialisation of a simulation at one moment in time is required. So, one seeks to get time independent variables. The intention of this paper is to give an idea of how the approach can be applied. Further analysis results that actually give guarantees are for a future publication.

4 Engineering based on Analysis Results

The biggest problem with engineering self-organising emergent systems is the lack of a systematic approach to build a solution that meets the requirements. Despite some efforts (e.g. [14,15]) to find a systematic engineering approach, finding an approach that starts from the macroscopic requirements and systematically constructs a system by deriving the behaviours of the individual agents in the system from the macroscopic requirements seems infeasible. Therefore, a combination of creatively building a solution and analysing it based on scientifically founded experimental methods is promising. Traditional engineering design methods tend to be based on a bottom-up approach in which known components are assembled into subsystems from which the system is constructed and then tested for the required properties. The design is modified in an iterative manner until the system meets the requirements. As discussed in [4,5], a formal design method often can not do the job and the authors argue that such a method does not exist. One needs an experimental scientifically founded method.

We propose an integration of the systematic analysis approach, which we described earlier, into the engineering process. First of all, as a creative activity, one builds a first prototype of the system based on experience and combining existing mechanisms (e.g. [16]) and guidelines (e.g. [15]) to achieve a self-organising emergent system. Then one systematically analyses the system with respect to the wanted macroscopic requirements using the analysis approach described above. Feedback from that analysis is then used in a next engineering cycle to adjust and tune the solution in order to systematically evolve towards a final solution that meets all the requirements. Of course, the feedback obtained through the analysis of self-organising emergent systems can also result in more experience and guidelines to use in future engineering processes.

Using the above analysis approach extensively in an engineering process gives a number of possibilities. Some examples are:

- Bifurcation Analysis based on Parameters: As explained in section 3.3, the analysis approach allows numerical analysis algorithms to directly steer the simulation process in order to obtain simulation results on the fly and as efficiently as possible. One such analysis algorithm is a so called bifurcation analysis. Based on a certain parameter and for a given macroscopic variable the algorithm analyses what the influence of that parameter is on the macroscopic behaviour, i.e. on the macroscopic variable. For example, in the AGV case the number of AGVs used could be an important parameter. The results can then indicate how many AGVs are ideal for a given system to meet its requirements as best as possible and what happens when the number of AGVs goes beyond the ideal range of values. Also, in the throughput example (see section 3.4) the arrival rate $AR_{\Delta t}$ is an interesting parameter in order to know for which utilisation load a certain solution performs within acceptable boundaries. Such results are useful feedback to systematically tune and re-engineer a solution for the AGV case study.
- Comparison and Evaluation of existing Decentralised Mechanisms: In today's self-organising emergent systems a number of decentralised mechanisms are used, of which a lot are inspired by nature [16] (e.g. pheromones, gradient fields, etc.). Evaluating each of the decentralised mechanisms with respect to for example non-functional and other characteristics (scalability, flexibility, reaction-speed to changes,

communication bandwidth used, etc.) allows a more precise comparison of those mechanisms. Also, each mechanism has a number of parameters one has to tune (e.g. the evaporation rate for digital pheromones). An analysis can be done of the influence of the parameters and in which context which parameter range is most suitable. Further, when engineering a self-organising emergent system, this information will guide the choice of appropriate mechanisms for the problem at hand and thus contribute to a more systematic engineering of such systems.

Exploiting such possibilities integrates the analysis approach into the engineering process such that a constant feedback loop between analysis and engineering shapes a self-organising emergent solution. In the end, a systematic simulation-based engineering process where scientific analysis and feedback are essential can be achieved.

5 Conclusion and Future Work

It is clear that the way to systematically build a self-organising emergent system remains an open issue. However, before even considering a self-organising emergent system in an industrial context, a systematic analysis approach is needed that gives guarantees that the required coherent macroscopic behaviour is achieved and maintained. Because formal or analytic proofs are infeasible and because mere observation of simulation results is not reliable and scientific enough, the proposed approach combines realistic agent-based simulations with existing scientific numerical analysis algorithms for dynamical systems. Compared to mere observation of simulation results, more reliable and valuable results are returned because the analysis algorithm itself obtains the results on the fly by steering and accelerating the simulation process according to the algorithm's goal. In order to also achieve a more systematic engineering approach, it was proposed to exploit the analysis approach during the engineering process such that a constant feedback loop between scientific analysis and engineering shapes a self-organising emergent solution.

Future work includes applying the approach extensively to the AGV case and other application domains. Also, there are a number of issues that need to be resolved in order to make the approach easy applicable. For example, there are no clear guidelines on how to chose suitable macroscopic variables that reflect the macroscopic properties under study. Next to making the analysis approach more straightforward, the integration of that analysis approach into a systematic engineering approach can also be worked out and be made explicit in for example an engineering road map. Such a road map would then indicate where in a traditional software engineering methodology the scientific analysis is situated and where feedback from that analysis is to be used to guide the engineering process.

Acknowledgements. This work is supported by the K.U.Leuven research council as part of the Concerted Research Action on Agents for Coordination and Control - AgCo2 (T. De Wolf,T. Holvoet). G. Samaey is a Research Assistant of the Fund for Scientific Research – Flanders (FWO). This work is supported by grants FWO G.0130.03, IUAP/V/22 (Belgian Federal Science Policy Office) (G. Samaey).

References

1. Egemin, DistriNet: Emc^2: Egemin Modular Controls Concept,. (IWT-funded project with participants: Egemin (http://www.egemin.be) and DistriNet (research group of K.U.Leuven)) Started on 1 March 2004, ending on 28 February 2006.
2. De Wolf, T., Holvoet, T.: Emergence and Self-Organisation: a statement of similarities and differences. In: Proceedings of the Second International Workshop on Engineering Self-Organising Applications, New York, USA (2004) 96–110
3. Wegner, P.: Why Interaction is More Powerful than Algorithms. Communications of the ACM **40** (1997) 80–91
4. Edmonds, B., Bryson, J.J.: The Insufficiency of Formal Design Methods - the necessity of an experimental approach for understanding and control of complex MAS. In: Proceedings of the 3rd Internation Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04), New York, ACM Press (2004) 938–945
5. Edmonds, B.: Using the Experimental Method to Produce Reliable Self-Organised Systems. In Brueckner, S., Serugendo, G.D.M., Karageorgos, A., Nagpal, R., eds.: Engineering Self Organising Sytems: Methodologies and Applications. Volume 3464 of Lecture Notes in Artificial Intelligence., Springer (2004) (to appear spring 2005).
6. Anthony, R.J.: Emergence: A Paradigm for Robust and Scalable Distributed Applications. In: Proceedings of IEEE International Conference on Autonomic Computing (ICAC'04), New York (2004) 132–139
7. Parunak, H.V.D., Savit, R., Riolo, R.L.: Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In: MABS. (1998) 10–25
8. Wilson, W.: Resolving Discrepancies between Deterministic Population Models and Individual-Based Simulations. American Naturalist **151** (1998) 116–134
9. Kevrekidis, I.G., Gear, C.W., Hummer, G.: Equation-free: The computer-assisted analysis of complex, multiscale systems. AIChE Journal **50** (2004) 1346 – 1355
10. Kevrekidis, I.G., Gear, C.W., Hyman, J.M., Kevrekidis, P.G., Runborg, O., Theodoropoulos, C.: Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. Communications in Mathematical Sciences **1** (2003) 715 – 762 (available online at http://www.intlpress.com/CMS/).
11. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C: The Art of Scientific Computing. 2nd edn. Cambridge University Press, Cambridge (1992)
12. Guerin, S., Kunkle, D.: Emergence of Constraint in Self-Organizing Systems. NDPLS: Nonlinear Dynamics, Psychology, and Life Sciences **8** (2004) 131
13. De Wolf, T., Samaey, G., Holvoet, T., Roose, D.: Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods. In: Proceedings of IEEE International Conference on Autonomic Computing (ICAC'05), Seattle, USA (2005) (accepted).
14. Poulton, G., Guo, Y., James, G., Valencia, P., Gerasimov, V., Li, J.: Directed Self-Assembly of 2-Dimensional Mesoblocks using Top-down/Bottom-up Design. In: Proceedings of the Second International Workshop on Engineering Self-Organising Applications (ESOA04), New York, USA (2004) 137–149
15. Parunak, H.V.D., Brueckner, S.A.: Engineering Swarming Systems. In Bergenti, F., Gleizes, M.P., Zambonelli, F., eds.: Methodologies and Software Engineering for Agent Systems. Volume 11 of Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer (2004)
16. Nagpal, R.: A Catalog of Biologically-inspired Primitives for Engineering Self-Organization. In Serugendo, G.D.M., Karageorgos, A., Rana, O.F., Zambonelli, F., eds.: Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering. Volume 2977 of Lecture Notes in Computer Science., Springer (2004) 53–62