Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

USING AGENT-BASED SIMULATION FOR EMERGENT BEHAVIOR DETECTION IN CYBER-PHYSICAL SYSTEMS

Rob Bemthuis
Martijn Mes
Maria-Eugenia Iacob
Paul Havinga

University of Twente
Drienerlolaan 5
Enschede, 7522 NB, THE NETHERLANDS

ABSTRACT

Traditional modeling approaches, based on predefined business logic, offer little support for today's complex environments. In this paper, we propose a conceptual agent-based simulation framework to help not only discover complex business processes but also to analyze and learn from emergent behavior arising in cyber-physical systems. Techniques originating from agent-based modeling as well as from the process mining discipline are used to reinforce agent-based decision-making. Whereas agent-technology is used to orchestrate the integration and relationship between the environment and business logic activities, process mining capabilities are mainly used to discover and analyze emergent behavior. Using a functional decomposition approach, we specified three agent types: cyber-physical controller agent, business rule management agent, and emergent behavior detection agent. We use agent-based simulation of a logistics cold chain case study to demonstrate the feasibility of our approach.

1 INTRODUCTION

The past decade has seen revolutionary changes in information exchange. Organizations, enterprises, and devices are increasingly interconnected and business environments are changing rapidly to advance their infrastructures such that they can cope with new or unforeseen sources of emergent behavior. Recently, the cyber physical system (CPS) paradigm has emerged as a promising direction to enrich interactions between physical objects and virtual worlds (Rajkumar et al. 2010). A CPS comprises of interconnected, yet autonomous, physical assets, and computational capabilities (Baheti and Gill 2011).

Exchanging and aligning the needs from data providers (e.g., sensors) on the one hand, and data users (e.g., decision makers) on the other hand, furnish a breeding ground of technological challenges in CPS. The diversity of data communication entities and the tendency to distribute large portions of the applications close to the point of where the data is being gathered pose risks of disruptive events. This is also reflected in today's decision-making: more and more tasks are delegated to distributed systems (Alcácer and Cruz-Machado 2019). A mechanism to create the right balance between central control and autonomy is required to properly address the risks of disruptive events and ensure resilience of CPS.

Although current mechanisms to manage business logic in CPS can act upon a wide range of possible disruptions, much published research on this topic may not generalize easily. In addition, comprehensive analysis methods can be hard to obtain or maintain due to prevailing (i) complexity of a CPS (e.g., many physical entities, conflicting interests, no coordination, etc.), (ii) discrepancies between expected behaviors and realized results, and (iii) difficulties with capturing and affecting emergent behavior in a CPS. Centralized approaches with predefined steps of activities may be inadequate for handling these complex and dynamic

978-1-7281-9499-8/20/$31.00 ©2020 IEEE

environments (Wang and Wang 2006). Another challenge is that before business logic can be modeled and managed, it first needs to be captured and extracted, which is often left to analyst's intuition (Levina et al. 2010). Furthermore, CPSs representing both physical and cyber parts require information at design-time to verify and validate the designed models but also at run-time for monitoring and reconfiguration (Graja et al. 2020), which imperatively requires a form of self-learning behavior. A promising way to handle these challenges is to focus on the individual agents of a system and the emerging behavior arising from the interactions between these agents. As there is not much published research available on modeling business logic in such a way, we decide to focus on this. More precisely, we present an agent-based simulation (ABS) model to show how a CPS could be enhanced such that it could properly handle disruptions.

The objective of this paper is to develop an ABS framework in which the interaction and relationship between a CPS environment and business logic activities can be observed and managed. Simulating this type of symbiosis can be an appropriate approach for determining how to coordinate business logic tasks and comprise it in CPSs. By embedding a gradual and iterative approach, we aim to create a feedback loop in which the agents can learn from discovered emergent behavior. In turn, to capture insights into emergence, we decide to use process mining techniques. The contribution of this work is twofold: (1) we propose an ABS framework to evaluate the planning and control of business logic utilizing process mining to capture the emergent behavior in CPSs, and (2) we show how our framework can be applied by demonstrating its applicability to a case study from the logistics domain.

The remainder of this paper is structured as follows. Section 2 provides background materials and contrasts related work. Section 3 presents the ABS framework. Section 4 discusses the logistics case study. Section 5 gives the experimental setup. Section 6 presents our model verification and validation. We end with conclusions and directions for further research in Section 7.

2 BACKGROUND AND LITERATURE REVIEW

This section provides background on agent-based models focusing on analyzing emergent behavior. Subsequently, this section gives an overview of related works on business logic and process mining.

It is well known that emergent behavior results from the interactions between individual agents. A classic example is the phenomenon of flocking birds: autonomous birds behave according to simple local rules, but as part of a community each bird is engaging in the community's self-organizing behavior, and flocking patterns emerge. We follow the definition of emergence as given by Grieves and Vickers (2017), which describes emergence as a result of unfamiliar and unique behavior that has occurred or as unrecognized behavior that has been a possibility from inception. Without disregarding all accounts of emergence and agent-based modeling, we will now address some approaches used to discover or analyze emergence for ABS models. Van der Zwet et al. (2019) propose an agent-based model to study how the interactions of individuals under different conditions lead to emergence. In a broader sense, Lee et al. (1998) study the performance and scalability of multi-agent applications, by considering different hierarchical relational structures. Although these two studies demonstrate how agent strategies can impact emergent behaviors, the decisions of the agents are not updated based on the resulting observed emergent phenomena. To highlight the potential, however, research of Choi et al. (2001) examines the trade-off between agent control and emergence. They underpin the importance of being able to not only control agent activities but also remain vigilant, observe what emerges, and make decisions appropriately, which is in line with our envisioned work. Further empirical evidence needs to be obtained to examine relationships that cause emergent phenomena of agent behavior. In this paper, we extend the typical ABS approach by embedding techniques and algorithms for emergent behavior analysis. We show how the feedback mechanism between agents, as conceptualized by Bemthuis et al. (2019), and emergent behavior analysis methods can be realized. We decide to focus on analyzing data from the environment and amplify this with an agent-based decision support system for CPSs.

Business logic describes how the business derives actions from conditions (Von Halle and Goldberg 2010) and transforms data to value. The evaluation of the conditions can be expressed in so-called business rules, which are statements and criteria used to guide behavior, shape judgments, or make decisions (Ross 2013). A business rule is under business jurisdiction, which means that a business can enact, revise, and discontinue their rules as they deem appropriate. The aforementioned operational constraints, exception handling, and business strategy may also play a role. Several studies put forward the idea of embedding the business logic in software systems close to the point where the data are collected using, e.g., mobile devices. For example, Zhao et al. (2012) incorporate business logic into RFID-enabled applications. Their model extracts events from RFID data and guides the behaviors and interactions of involved objects in response to the extracted events and a set of pre-defined business rules. Levina et al. (2010) suggest a method to elicit business rules from business process models. In the same vein of these works, we provide a means to incorporate business logic at a cyber-physical level, and at the same time update decision-making as well, by evaluating emergent behavior.

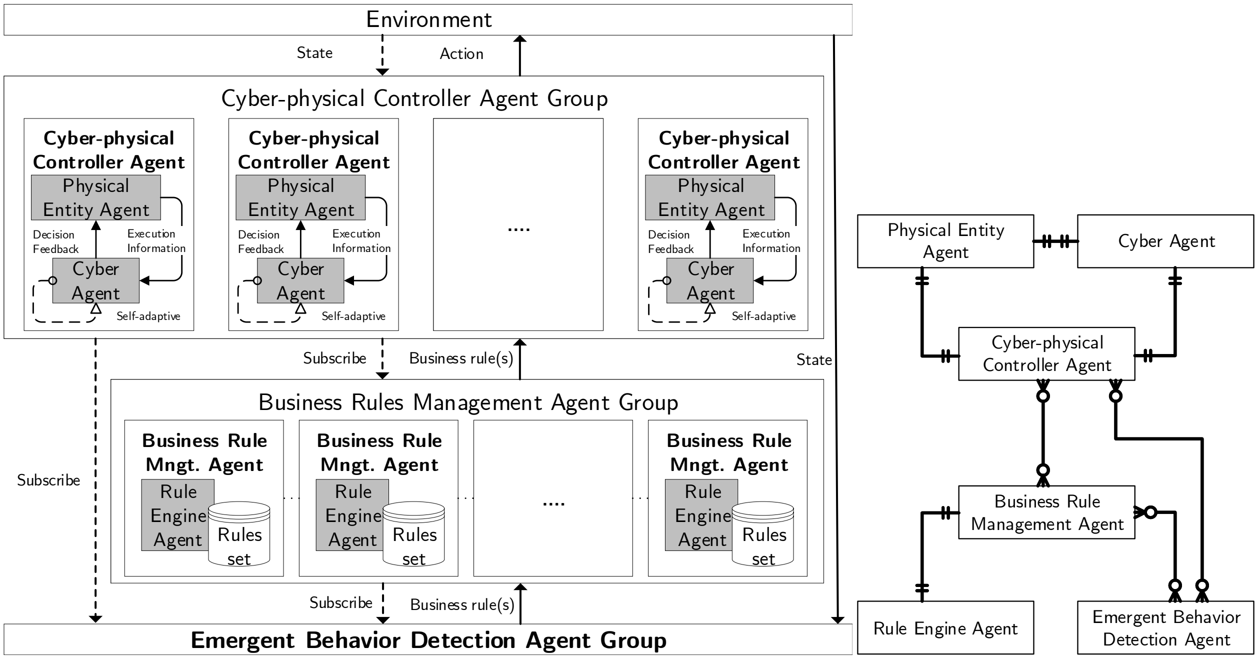
The goal of process mining is to discover, monitor, and analyze processes by extracting process knowledge and process relation information from event logs (Van der Aalst 2011). Event logs consist of business events described by at least three attributes: case, activity, and timestamp. The case, or process instance, refers to the event's object that undergoes the process (e.g., a customer order). An activity represents a well-defined step in the business process (e.g., pick-up customer order). The sequence of events for a particular case is called a trace. A timestamp denotes the moment the activity is executed. Additionally, the event log can store other detailed information on events concerning, for example, the resource involved (e.g., a person or machine) and transactional data (e.g., costs). From the traces, process models can be generated, which can be analyzed from various perspectives (Van der Aalst et al. 2011). Even though some research on modeling approaches for rapidly changing environments are available regarding process mining, the level of coping with CPSs is still limited. Augusto et al. (2016) propose a methodology to perform analysis based on raw database records using process mining techniques. They use an ABS model for validation of their work. Martin et al. (2014) consider the use of process mining in a simulation context. They foresee the potential of process mining for retrieving insights on which activities are executed under which circumstances. Hence, process mining could be used to discover decision logic. Lau et al. (2009) propose a process mining system for knowledge discovery in daily operational logistics data. Although their model is able to identify process instances that relate to the level of customer satisfaction, there is limited feedback from the discovered models to improve the overall system performance. Moreover, the model focuses on process discovery, while it would be interesting to incorporate a form of self-learning capabilities as well. As illustrated in our previous work (Bemthuis et al. 2019), process mining methods are useful to improve the effectiveness and efficiency of business processes by analyzing emergence.

3 MODEL FRAMEWORK

A CPS consists of two main functional components: (1) advanced sensing and connectivity ensuring real-time data acquisition from the physical world and feedback from the cyber space; and (2) intelligent data management, analytics, and some computational capability (Lee et al. 2015). In general, these components define the relational constructs of a CPS. Figure 1 presents an overview of our framework, including simplified interactions between the agents and the environment. In the remainder of this section, we outline the main functionalities of our framework, as decomposed in functional agents.

3.1 Cyber-physical Controller Agent

The Cyber-physical Controller Agent (CPCA) maintains the control of the business logic distributed at the level of a physical entity. It consists of a Physical Entity Agent (PEA) and a Cyber Agent (CA). The PEA manifests itself as a smart connection interface, responsible for acquiring data about the environment as measured by sensors or obtained from a controller or enterprise manufacturing system. PEA's purpose is also to act as an actuator and carry out commands that are given by the CPCA. The CA determines the decisions made at the physical object, and relies upon information gathered to provide insight into the status


Environment
State
Action
Cyber-physical Controller Agent Group
Cyber-physical Controller Agent
Physical Entity Agent
Decision Feedback
Execution Information
Cyber Agent
Self-adaptive
....
Subscribe
Business rule(s)
State
Business Rules Management Agent Group
Business Rule Mngt. Agent
Rule Engine Agent
Rules set
Subscribe
Emergent Behavior Detection Agent Group
Physical Entity Agent
Cyber Agent
Cyber-physical Controller Agent
Business Rule Management Agent
Rule Engine Agent
Emergent Behavior Detection Agent

(a) Interactions for all agent types.

(b) Entity relationship diagram.

Figure 1: A conceptual agent-based framework for business logic and emergent behavior analysis in cyber-physical systems.

of the individual physical entity (e.g., machine). The CPCA is collaborating with other CPCAs as well as with the Business Rule Management Agents (BRMAs). In turn, a CA can extract information from these sources to enrich its self-learning capabilities. It can decide by itself how to evaluate the conditions to arrive at a conclusion (the business logic). The two functional CPS components are linked by the feedback loops where collaborating computational entities (embedded in agents) are connected (i.e., monitored, controlled, coordinated, and integrated) with the surrounding physical world and its on-going processes.

3.2 Busines Rule Management Agent

The CPCA can be connected with so-called Business Rule Management Agents (BRMAs), which can help to derive conclusions from a set of conditions (see Figure 1a). A BRMA manages a collection of business rules as determined by an organization or originated/updated via the Emergent Behavior Detection Agent (EBDA). We explain more about the EBDA in the next subsection, for the explanation of BRMA's functionalities below it suffices to think about EBDA as a means to extract business rules from ubiquitous sources such as websites, humans, etc. The synergetic interaction between CPCAs and BRMAs is one of the key components of our presented framework (see Figure 1b). A BRMA can push new or updated business rules to a CPCA but a CPCA can also request or subscribe to new/updated business rules from a BRMA. A mechanism to control this can be a publish-subscribe procedure, where agents can subscribe themselves to receive updates about business rules of interest. In turn, each CPCA has to decide upon which business rules to adhere to, and what action to undertake. These actions can be to call for new/updated business rules, to push a new business rule, or to simply execute a business strategy according to the business logic.

A multitude of relationship typologies can be used for describing the interaction between CPCAs and BRMAs. The agent instances as shown in Figure 1b may actually take the form of different configurations, as those exemplified in Figure 2. These relations define how the CPCAs and BRMAs may interact with one another. Note that the interactions among agents of a similar type are not shown in this figure. In a dedicated relation, each CPCA has its own dedicated BRMA. The BRMA focuses on orchestrating the needs for business rules from/to the CPCA and the EBDA. In a distributed relation, a CPCA can be connected with one or more BRMAs. The centralized relationship assumes that we have some control over a group of CPCAs and BRMAs that may work together. That is, instead of having the CPCAs decide by themselves how and when to communicate with a BRMA, the orchestration mechanism decides how to coordinate the exchange and update of business rules based on requests and offers from its users (e.g., CPCAs, BRMAs). The goal of the central authority can be to achieve an efficient allocation of business rules. Hierarchical, mixed, and other forms of relational constructs are also possible because of the (mutual) agent interaction.

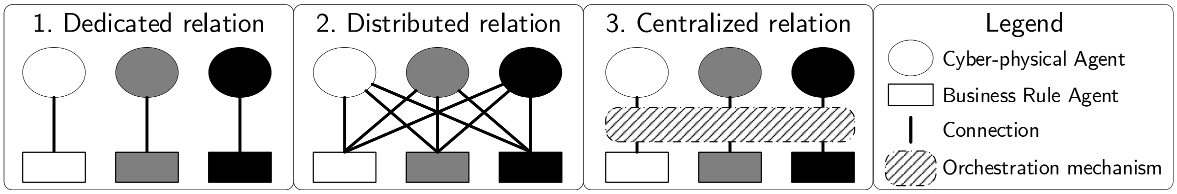


Figure 2: Relational strategies for connecting cyber-physical agents with business rule agents.

3.3 Emergent Behavior Detection Agent

The goal of this agent is twofold: to adsorb data about the environment and to provide updates about business rules. To this end, it uses techniques for analyzing emergent behavior from the environment and from the BRMAs and CPCAs. A BRMA or CPCA may namely request updates about special cases of emergent behavior, for example, about some business rules that are not captured by its current ruleset. In principle, the EBDA is a special kind of agent that focuses on detecting (and, more broadly, analyzing) emergence, which is the accumulated result of agent interactions and other environmental behaviors.

As an example of emergent behavior arising from the environment, suppose the government decides to make certain food regulations stricter. This can mean that a new regulation may require that the temperature of logistics goods should be monitored every minute, and data about abrupt deviations should be stored. The EBDA extracts this information and converts this into new or updated business rules. This could be done through, for example, analyzing streams of event logs. Individual BRMAs get then aware of these new/updated business rules. For example, one BRMA responsible for fresh fish now decides to set the temperature sample rate to 30 seconds and another BRMA decides to adjust its data storage policy. As another example, suppose a CPCA that wants to check whether there is an update about a business rule. Consider having a CPCA responsible for a vehicle. The CPCA decides to follow the business rule that prescribes to go left at a certain junction. A moment later, another CPCA needs to make a similar decision. However, for some reason, this agent is doubting whether going left is the best decision. The CPCA, therefore, requests the BRMA to check for updates regarding the associated business logic. In turn, the BRMA requests an update from the EBDA. The BRMA also wants to check with the EBDA whether there are some new insights to be gained from emergent behaviors. The EBDA assesses the emergent behavior, for example concerning the impact of the other CPCA that decided to go left, and informs the BRMA that the vehicle should take the right path because that is expected to be more beneficial. These examples show that EBDA plays a prominent role in capturing knowledge about the environment and incorporating this knowledge dynamically into actual business rules.

The proposed framework not only allows to capture emergent behavior from real-world CPS but is also able to introduce or manipulate the emerging behavior by changing the agents' decisions. In turn, the EBDAs can assess the validity and impact of the strategies employed by the individual agents on the overall system. Furthermore, a similar relationship can be established between the EBDA and BRMA as exemplified with the CPCA and the BRMA (Figure 2). Having one EBDA may be seen as having one single algorithm that has full control and overview of the emergent behavior. For example, a process mining algorithm can be used for that purpose but also a human can fulfill this role. In a distributed setting, EBDAs have their own perception of emergent behavior and means to absorb data, convert it to business rules, and provides updates to the BRMAs. For example, this could be implemented by having multiple process mining algorithms, each responsible for a class of emergent behaviors.

4 CASE STUDY

This section introduces our logistics case study. First, we briefly outline the case, after which we describe two components in more detail, the facility layout and the fleet and product composition.

4.1 Internal Transport of Smart Pallets in a Production Facility

The diversity of channel partners (e.g., suppliers, intermediaries, third party service providers, customers, etc.), heterogeneity of data sources, and advancements in IT infrastructures motivated us and our project partners to conduct a case study for an internal transportation network. The case study is about a production facility, which is data-intensive and uses consolidation approaches to bundle multiple sources of data for decision-making. With the vast influx of data in a connected environment, decision-making entities (e.g., forklift drivers, facility managers, etc.) are expected to receive relevant information and to adequately respond in a timely manner to, e.g., emerging disruptions.

The case study involves the transport of products that are subject to quality depreciation. Pivotal in this case study is the role of smart pallets, allowing to monitor the integrity of the shipments. The pallets are equipped with sensors and communication means, which enables us to identify and locate a pallet, but also to keep track of the speed, vibrations (e.g., shocks), temperature, humidity, pollution (e.g., CO2), and weight. The quality of the logistics goods we consider may be affected during storage and transport. By utilizing knowledge obtained from the smart pallets, conditions about the state of the logistics goods can be derived in a real-time manner. Moreover, having computational capabilities attached at the physical entity, makes this case a promising candidate for validating our framework.

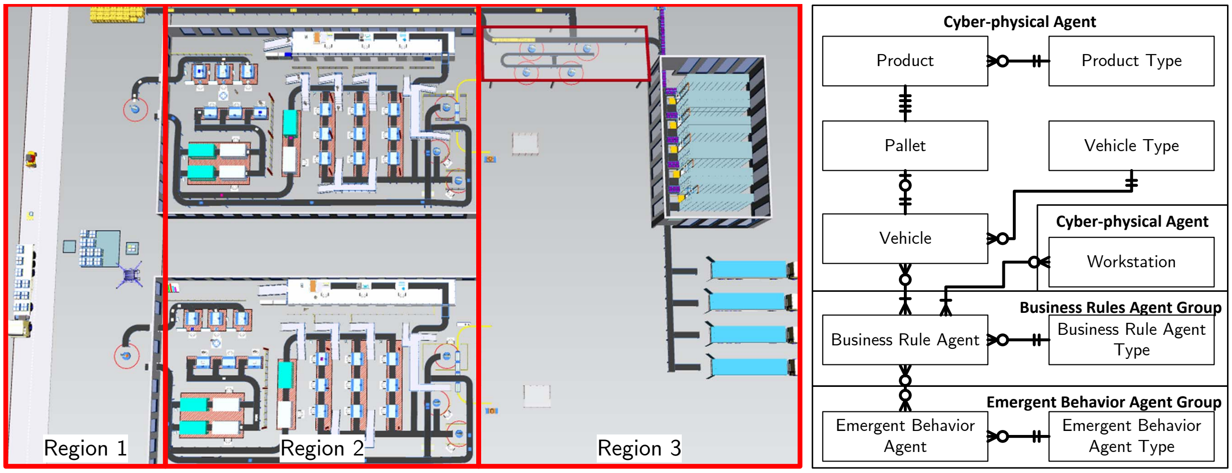
More specifically, we consider the problem as a special case of the so-called Dynamic Pickup and Delivery Problem (DPDP). DPDPs belong to the class of vehicle routing problems in which objects need to be transported between an origin and a destination (Berbeglia et al. 2010). In the dynamic variant, input data are revealed over time. A solution for this problem is described by a strategy, specifying which actions need to be performed as time passes (Berbeglia et al. 2010). When transporting the goods, minimizing quality decay must be balanced against minimizing operating costs. Together with the (industrial) project partners, we incorporated some practical requirements at this design stage (as discussed hereafter), with the aim of closely representing reality.

4.2 Facility layout

The considered production facility focuses on designing transport routes and schedules for internal cargo transporters. Logistics goods are specified by pickup and delivery requests between origins and destinations. In the plant layout (see Figure 3a), there are three regions specified. The first region facilitates transport from a conveyor belt to a climate-controlled environment. In the second region, the products are processed, packed, and prepared for shipment. The third region focuses on transporting the pallets from a pick-up area to climate-controlled trucks. Region one and three are the focus areas of this study, the second region is namely operated by an external party. It is also not possible to transit vehicles from one region to another region within the plant.

4.3 Fleet Composition and Products Facilitated

The case study involves a heterogeneous fleet of cargo transporters with fixed capacities. Three types of transporters are included: unmanned aerial vehicles (UAV), human-driven forklifts (HDF), and automated



(a) Plant layout (made in Tecnomatix Plant Simulation v14). (b) Entity relationship diagram.

Figure 3: High-level schematic overview of the case study.

guided vehicles (AGV). As shown in Table 1, each vehicle has its own properties. The vehicles operate on separated dedicated lanes, of which we assume that they do not interfere with each other. Among these properties, the quality decay factor requires some more attention. We assume that the quality of the goods transported is subject to deterioration, according to a certain rate as expressed with an exponential decay function: $N(t) = N_0 e^{-\eta t}$, where $N(t)$ is the quantity at time t and η an exponential decay constant.

Likewise, the products are characterized by certain properties (see Table 1). To obtain the quality decay rate of a vehicle during transport of a product, we imply that one could multiply the vehicle decay function with the corresponding product decay function. Since the focus of our study is not on region 2, we simply assume that there is no quality decay during that period. We consider that transportation requests are gradually revealed by means of a Poisson arrival process, and assume product-dependent exponentially distributed processing times in region 2. Furthermore, we assume that certain product types may not be transported together and that buffer sizes are unrestricted.

Table 1: Vehicle and product characteristics.

Vehicles				Products				
Property	UAV	HDF	AGV	Property	Product 1	Product 2	Product 3	Product 4
Speed (km/h)	12.0	6.0	4.0	Size (units)	Small (1)	Medium (2)	Medium (2)	Large (3)
Capacity (units)	1	3	10	May not be transported together	-	Product 3	Product 2	-
Number available	6	2	2	Processing time (minutes)	~ N(3.0,0.5)	~ N(4.0,0.40)	~ N(4.0,0.40)	~ N(5.0,0.60)
Costs per vehicle per hour	100	200	500	Quality decay constant (η) at region 1	$5.0 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
Quality decay constant (η)	$2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$5.0 \cdot 10^{-3}$	Quality decay constant (η) at region 3	$4.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
Distance region 2 & 3 (meters)	75 & 100	90 & 120	120 & 150	Mean interarrival time (seconds) ($1/\lambda$)	10	20	20	30

5 Simulation Modeling

To validate the framework, we evaluate various agent configurations. Below, we first give an overview of the ABS model, after which we discuss the modeling choices regarding the CPCA, BRMA, and EBDA. We implemented our model in the simulation software Tecnomatix Plant Simulation. Figure 4 shows a screenshot of the simulation model.

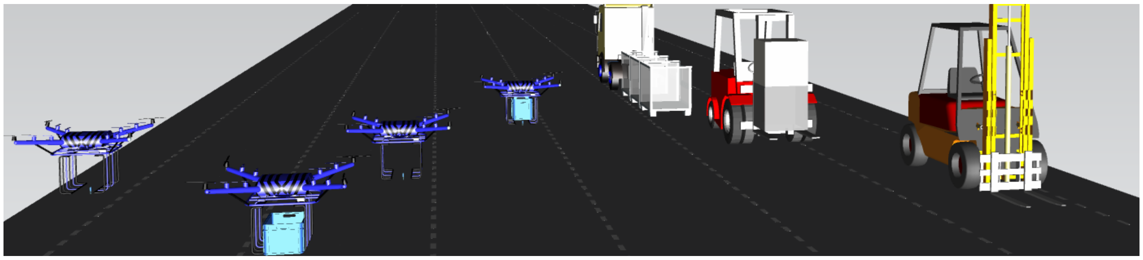


Figure 4: Snapshot of the simulation model.

5.1 Modeling the Cyber-physical Controller Agent

CPCAs act as physical assets with the capabilities of making autonomous decisions. We model three types of CPCAs: vehicle CPCAs, pallet CPCAs, and workstation CPCAs. The vehicle CPCA decides upon actions carried out by the vehicle. This decision is based on the state as determined by the physical part of the vehicle (e.g., capacity, position, etc.) and the set of business rules. CPCAs adhere to the rules as discussed in the business rule agent modeling part (see below). A pallet CPCA is responsible for managing cyber-physical tasks related to the products and pallets. That is, initiating transport tasks and keeping track of the quality decay. The workstation CPCA is responsible for actions carried out before or after a vehicle CPCA picked-up or respectively dropped-off the pallet(s), such as processing the products.

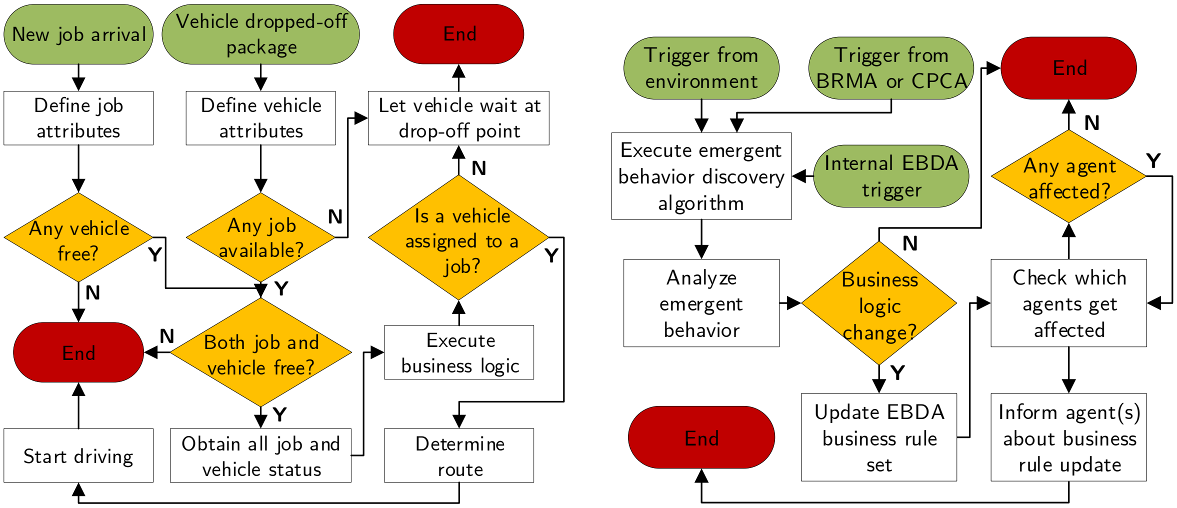
The modeling for the CPCAs is based on a few simplifications and assumptions. Although some of the vehicles could transport the cargo both horizontally and vertically, we simply consider the travel distances as given in Table 1 to be leading. Thereby, the given speed includes potential corrections for (de-)acceleration. Furthermore, the vehicle controller responsible for the physical movements of the vehicles is outside the scope of this study. For that reason, we consider the communication on a physical component level to only communicate driving actions to the vehicle controller. Also, we assume that the vehicle's controllers solve potential collision avoidance maneuvers or conflicts. Likewise, the management of recharging, refueling or maintenance strategies is not part of our study. Instead, we assume there is sufficient time during parking to fulfill these kinds of needs and that additional compensation is embedded into the cost composition. The physical size of the vehicles is also not relevant in our model because we consider that the decision-making moments are based on the central midpoint of the vehicle. In practice, one could correct this at the controller level. Lastly, the pick-up and drop-off locations provide enough space to hold all vehicles and pallets.

5.2 Modeling the Business Rule Management Agent

For our case study, we decide to focus on the planning and control of the vehicles by using relatively simplistic dispatching rules. The dispatching rule is an algorithm for assigning jobs to vehicles or vice versa, and can be seen as a special case of a business rule, which seems appropriate for modeling our BRMAs. We further decide to dedicate one BRMA to each vehicle type, product type, and workstation. Considering decoupling and cohesion criteria, we use three types of vehicle BRMAs, four types of pallet BRMAs and one type of workstation CPCA. These BRMAs each have a one-to-one relationship with their CPCA counterpart. For the sake of clarity, we depict the logic according to the studied business rules (as defined hereafter) as shown in Figure 5a.

5.3 Modeling the Emergent Behavior Detection Agent

Event logs captured or executed by individual agents are used as input for the emergent behavior discovery algorithm. The aggregated event traces are represented in the form of mined process models. The process of deducing emergence (e.g., gaining insights from the mined process models) through traces of events, is done by the emergent behavior discovery algorithm and the subsequent analysis part (see Figure 5b). For the purpose of this study, we decide to implement only one EBDA, by using the process mining discovery


New job arrival
Vehicle dropped-off package
End
Define job attributes
Define vehicle attributes
Let vehicle wait at drop-off point
N
Any vehicle free?
Y
Any job available?
N
Is a vehicle assigned to a job?
Y
Y
N
End
N
Both job and vehicle free?
Execute business logic
Y
Start driving
Obtain all job and vehicle status
Determine route
Trigger from environment
Trigger from BRMA or CPCA
End
N
Execute emergent behavior discovery algorithm
Internal EBDA trigger
Any agent affected?
Y
N
Analyze emergent behavior
Business logic change?
Check which agents get affected
Y
End
Update EBDA business rule set
Inform agent(s) about business rule update

(a) Flowchart for job and vehicle arrivals.

(b) Flowchart for emergent behavior triggers.

Figure 5: High-level overview of the simulation model.

algorithm. As shown in Figure 5b, three triggers are used for executing the algorithm: environment, BRMAs or CPCAs, and the EBDA itself. In our analysis, we illustrate how business rules could be derived from the mined process models and, consequently, adapt the CPCAs' decisions.

6 VERIFICATION AND VALIDATION

Model verification and validation are important steps for establishing a sound model and for providing credibility to the model's user. Amongst others, the following verification techniques, as listed by Law (2015), have been used: debugging modules or subprograms, using a common simulation package (Tecnomatix Plant Simulation v14), running the model under simplifying assumptions, running the model under a variety of settings, and trace outputs. In addition, we used programming concepts such as using preconditions, postconditions, and invariants (e.g., a vehicle must not have a negative load) when reasoning about the correctness of the model. Furthermore, we used the implemented process mining algorithms to verify the behavior of the agents executed by the simulated processes, as inspired by Bemthuis et al. (2019). Finally, we used several black-box validation techniques (i.e., relating inputs to outputs of the system and addressing functionalities and requirements with a focus on the overall system behavior) and white-box validation techniques (i.e., validating constituent parts of the model). Based on these validation efforts, as well as on discussions with our industrial partner, we conclude that our ABS model is sufficiently accurate for the purpose of this study.

6.1 Experimental Configuration

The experiments we consider are based on business rules. Our first set of rules comprises dispatching rules, whereby we distinguish between product-initiated and vehicle-initiated dispatching rules. The product-initiated rules are: random (R), lowest utilization (L), and shortest travel distance (S). The vehicle-initiated rules are: random (R), lowest quality decay (L), and highest quality decay (H). Another rule we introduce is that products of which the quality has dropped below 60% (before being assigned in region 1), will be disposed. Scenarios with different combinations of business rules and number of vehicles are simulated.

6.2 Experimentation and Results

We analyzed two components of the framework in more detail. First, the effect of selected business rules as executed by the CPCAs, illustrating the effectiveness of our agent-based approach. Next, the extraction of new business rules through the EBDA. In the latter analysis, we demonstrate the agent's learning principle.

6.2.1 Impact of Business Rules on CPCAs

We conducted simulation experiments for three scenarios. Scenario 1 comprises 3 UAVs, 1 HDF, and 1 AGV in both region 1 and region 3. Scenario 2 includes 3 UAVs, 2 HDFs, and 2 AGVs in both regions. Scenario 3 is similar to scenario 2, but without UAVs. After applying Welch's method (Law 2015), we identified a warm-up period of 120 products (using the product cycle time as indicator). In the first considered experiment, this was obtained after roughly 15 minutes of simulation time, yet we decided to use a warm-up period of 2 hours. For this purpose, we use a non-terminating simulation, in which each run is executed for 11 hours. We omitted the first two hours and the last hour (the latter one for statistical data collection purposes). After conducting 20 runs per experiment, we obtained the results as shown in Figure 6. The figure shows the average cycle time (CT) per product and average decay level (DL) reached when a product leaves the system for the three scenarios. The experiment letters denotes respectively the aforementioned vehicle-initiated rule (R,L,H) and the product-initiated rule (R,L,S). The figure reveals that there is a link between the CT and the decline of DL, which is in line with expectations because the longer a product is in the system, the more time passes for decaying a product's quality. Interestingly, scenario 3 scores for most of the experiments the lowest DL and highest CT. There are some exceptions, such as with experiment HR, in which scenario 1 yields a higher CT and DL. Besides these differences between scenarios, the numbers also show that the choice of dispatching rules is important.

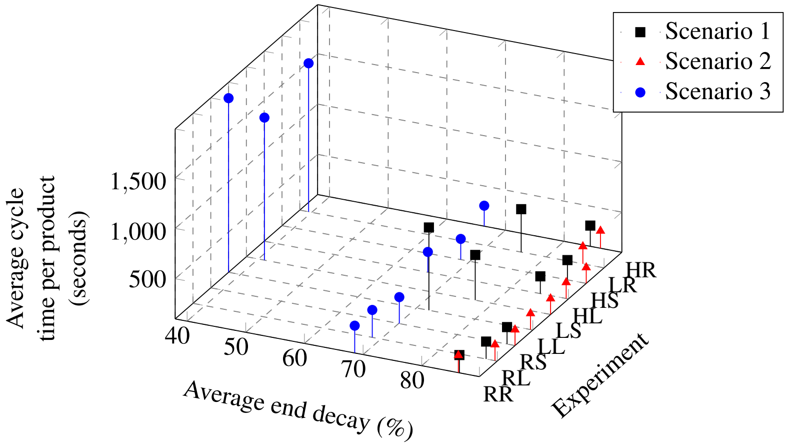
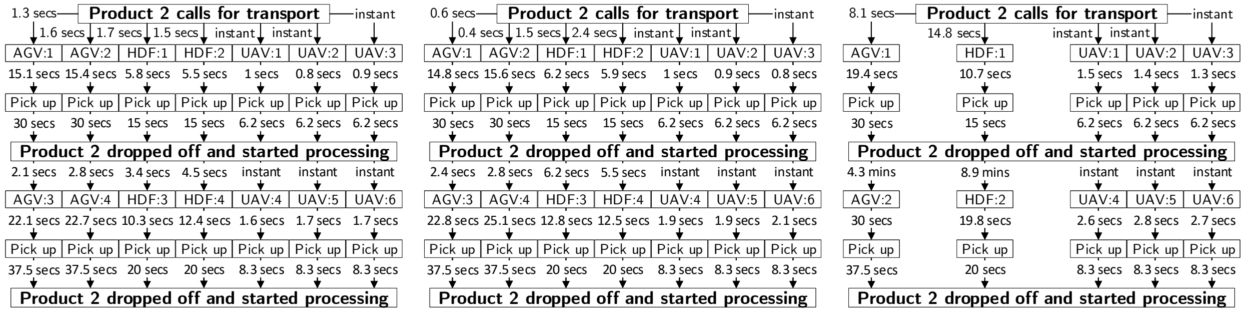


Figure 6: Simulation results for three scenarios.

6.2.2 Adapt Agent Decisions Based on Emergent Behavior Analysis

We show a basic form of agent's self-learning capabilities by conducting a few manual steps of how agent decisions can be adapted based on the mined process models. As more research should be carried out on that regard (Bemthuis et al. 2019), we believe that a promising first step is to demonstrate a few iterations. As starting point, let us consider the first run of scenario 2, experiment RR. For illustrative purposes, we assume that we do not know the results of the simulation study (Figure 6). Instead, we used the process mining software Disco and generated the process models using a fuzzy miner. Based on the first considered mined process models (a subset is shown in Figure 7a) the decision is taken to implement the lowest quality decay vehicle-initiated rule. Applying the fuzzy miner algorithm to the newly obtained event logs, the (sample) process model as shown in Figure 7b is obtained. Again, based on the mined process models, it is decided that changes are necessary. To cut costs, the decision is taken to remove one UAV and one HDF per region. Based on the resulting mined models (see Figure 7c) the agent could now consider further steps. Studying the impact of agent decisions as demonstrated with process mined models looks promising and more research is required to further quantify agent-learning methods.


1.3 secs
Product 2 calls for transport
instant
1.6 secs 1.7 secs 1.5 secs instant instant
AGV:1 AGV:2 HDF:1 HDF:2 UAV:1 UAV:2 UAV:3
15.1 secs 15.4 secs 5.8 secs 5.5 secs 1 secs 0.8 secs 0.9 secs
Pick up Pick up Pick up Pick up Pick up Pick up Pick up
30 secs 30 secs 15 secs 15 secs 6.2 secs 6.2 secs 6.2 secs
Product 2 dropped off and started processing
2.1 secs 2.8 secs 3.4 secs 4.5 secs instant instant instant
AGV:3 AGV:4 HDF:3 HDF:4 UAV:4 UAV:5 UAV:6
22.1 secs 22.7 secs 10.3 secs 12.4 secs 1.6 secs 1.7 secs 1.7 secs
Pick up Pick up Pick up Pick up Pick up Pick up Pick up
37.5 secs 37.5 secs 20 secs 20 secs 8.3 secs 8.3 secs 8.3 secs
Product 2 dropped off and started processing
0.6 secs
Product 2 calls for transport
instant
0.4 secs 1.5 secs 2.4 secs instant instant
AGV:1 AGV:2 HDF:1 HDF:2 UAV:1 UAV:2 UAV:3
14.8 secs 15.6 secs 6.2 secs 5.9 secs 1 secs 0.9 secs 0.8 secs
Pick up Pick up Pick up Pick up Pick up Pick up Pick up
30 secs 30 secs 15 secs 15 secs 6.2 secs 6.2 secs 6.2 secs
Product 2 dropped off and started processing
2.4 secs 2.8 secs 6.2 secs 5.5 secs instant instant instant
AGV:3 AGV:4 HDF:3 HDF:4 UAV:4 UAV:5 UAV:6
22.8 secs 25.1 secs 12.8 secs 12.5 secs 1.9 secs 1.9 secs 2.1 secs
Pick up Pick up Pick up Pick up Pick up Pick up Pick up
37.5 secs 37.5 secs 20 secs 20 secs 8.3 secs 8.3 secs 8.3 secs
Product 2 dropped off and started processing
8.1 secs
Product 2 calls for transport
instant
14.8 secs instant instant
AGV:1 HDF:1 UAV:1 UAV:2 UAV:3
19.4 secs 10.7 secs 1.5 secs 1.4 secs 1.3 secs
Pick up Pick up Pick up Pick up Pick up
30 secs 15 secs 6.2 secs 6.2 secs 6.2 secs
Product 2 dropped off and started processing
4.3 mins 8.9 mins instant instant instant
AGV:2 HDF:2 UAV:4 UAV:5 UAV:6
30 secs 19.8 secs 2.6 secs 2.8 secs 2.7 secs
Pick up Pick up Pick up Pick up Pick up
37.5 secs 20 secs 8.3 secs 8.3 secs 8.3 secs
Product 2 dropped off and started processing

(a) Scenario 2 experiment RR run 1. (b) Scenario 2 experiment LR run 2. (c) Scenario 1 experiment LR run 3.

Figure 7: Part of mined process models for three manual iterations.

7 CONCLUSIONS AND FURTHER RESEARCH

This paper proposes an ABS framework that will aid decision makers in capturing and analyzing emergent behavior in CPS. We have undergone a rethinking of the problem by encapsulating business logic and drawing on a synthesis of ABS and emergent behavior analysis techniques in a CPS-context. A cold-chain logistics case study based on a DPDP is presented for demonstrating the working of the conceptual framework, whereby process mining techniques are used to capture emergent behavior. Our results suggest that the way business logic is incorporated in CPS affects how the system performs and how and what behavior emerges. Our approach also paves the way for more sophisticated analytical methods where agents can learn from their environment thereby improving their decision-making capabilities. The feasibility of our approach could be also be further explored for optimal decision-making.

ACKNOWLEDGMENTS

This work was supported by the Netherlands Organization for Scientific Research (NWO) under the DataRel project (grant 628.009.015), and EFRO, OP Oost program under the Countdown project.

REFERENCES

Alcácer, V., and V. Cruz-Machado. 2019. “Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems”. *Engineering Science and Technology, an International Journal* 22(3):899 – 919.

Augusto, V., X. Xie, M. Prodel, B. Jouaneton, and L. Lamarsalle. 2016. “Evaluation of Discovered Clinical Pathways Using Process Mining and Joint Agent-Based Discrete-event Simulation”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by P. Frazier, T. M. Roeder, R. Szechtman, and E. Zhou, 2135–2146. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Baheti, R., and H. Gill. 2011. “Cyber-Physical Systems”. *The Impact of Control Technology*:161–166.

Bemthuis, R. H., M. Koot, M. R. K. Mes, F. A. Bukhsh, M. Iacob, and N. Meratnia. 2019. “An Agent-Based Process Mining Architecture for Emergent Behavior Analysis”. In *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, October 28th–31st, Paris, France, 54–64.

Berbeglia, G., J.-F. Cordeau, and G. Laporte. 2010. “Dynamic Pickup and Delivery Problems”. *European Journal of Operational Research* 202(1):8–15.

Choi, T. Y., K. J. Dooley, and M. Rungtusanatham. 2001. "Supply Networks and Complex Adaptive Systems: Control Versus Emergence". *Journal of Operations Management* 19(3):351–366.

Graja, I., S. Kallel, N. Guermouche, S. Cheikhrouhou, and A. Hadj Kacem. 2020. "A Comprehensive Survey on Modeling of Cyber-Physical Systems". *Concurrency and Computation: Practice and Experience* 32(15):e4850.

Grieves, M., and J. Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In *Transdisciplinary Perspectives on Complex Systems*, edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves, 85–113. Cham: Springer.

Lau, H. C., G. T. Ho, Y. Zhao, and N. Chung. 2009. "Development of a Process Mining System for Supporting Knowledge Discovery in a Supply Chain Network". *International Journal of Production Economics* 122(1):176–187.

Law, A. M. 2015. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill, Inc.

Lee, J., B. Bagheri, and H.-A. Kao. 2015. "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems". *Manufacturing Letters* 3:18–23.

Lee, L. C., H. S. Nwana, D. T. Ndumu, and P. De Wilde. 1998. "The Stability, Scalability and Performance of Multi-Agent Systems". *BT Technology Journal* 16(3):94–103.

Levina, O., O. Holschke, and J. Rake-Revelant. 2010. "Extracting Business Logic from Business Process Models". In *Proceedings of 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, April 16th–18th, Chengdu, China, 289–293.

Martin, N., B. Depaire, and A. Caris. 2014. "The Use of Process Mining in a Business Process Simulation Context: Overview and Challenges". In *Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining*, December 9th–12th, Orlando, United States, 381–388.

Rajkumar, R., I. Lee, L. Sha, and J. Stankovic. 2010. "Cyber-Physical Systems: The Next Computing Revolution". In *Design Automation Conference*, June 13th–18th, Anaheim, United States, 731–736.

Ross, R. G. 2013. *Business Rule Concepts: Getting to the Point of Knowledge*. 4 ed. Business Rule Solutions, LLC.

Van der Aalst, W., A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van den Brand, R. Brandtjen, J. Buijs et al. 2011. "Process Mining Manifesto". In *International Conference on Business Process Management*, edited by F. Daniel, K. Barkaoui, and S. Dustdar, 169–194. Berlin, Heidelberg: Springer.

Van der Aalst, W. M. P. 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin, Heidelberg: Springer.

Van der Zwet, K., A. I. Barros, T. M. van Engers, and B. van der Vecht. 2019. "An Agent-Based Model for Emergent Opponent Behavior". In *Computational Science – ICCS 2019*, edited by J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, and P. M. Sloot, 290–303. Cham: Springer.

Von Halle, B., and L. Goldberg. 2010. *The Decision Model: A Business Logic Framework Linking Business and Technology*. Boca Raton, United States: Auerbach Publications.

Wang, M., and H. Wang. 2006. "From Process Logic to Business Logic — A Cognitive Approach to Business Process Management". *Information & Management* 43(2):179–193.

Zhao, X., C. Liu, and T. Lin. 2012. "Incorporating Business Logics Into RFID-Enabled Applications". *Information Processing & Management* 48(1):47–62.

AUTHOR BIOGRAPHIES

ROB H. BEMTHUIS is PhD Candidate within the Pervasive Systems research group at the University of Twente, The Netherlands. His email address is r.h.bemthuis@utwente.nl.

MARTIJN R.K. MES is Associate Professor within the department of Industrial Engineering and Business Information Systems at the University of Twente, The Netherlands. His email address is m.r.k.mes@utwente.nl. His research interests are transportation, multi-agent systems, stochastic optimization, discrete event simulation, and simulation optimization.

MARIA-EUGENIA IACOB is Professor in the Industrial Engineering and Business Information Systems department at the University of Twente, The Netherlands. Her email address is m.e.iacob@utwente.nl. Her research is focusing on enterprise engineering topics, such as: enterprise architectures, business rules, supply chain management and integration, service-oriented architectures, industry 4.0, and smart logistics. She is co-author of the ArchiMate language (and International Standard specification) for enterprise architecture modelling.

PAUL J.M. HAVINGA is Full Professor and chair of the Pervasive Systems research group at the University of Twente, The Netherlands. His email address is p.j.m.havinga@utwente.nl. His research interests include wireless sensor networks, large-scale distributed systems, and energy-efficient wireless communication.