TOWARDS HOLISTIC MODELING AND SIMULATION OF DISCRETE EVENT AND INDIVIDUAL BASED BEHAVIOR

Nikolaus Furian
University of Auckland
Department of Engineering Science
70 Symonds Street, 1010 Auckland
New Zealand
E-mail: nfur003@aucklanduni.ac.nz

Dietmar Neubacher and Siegfried Vössner
Graz University of Technology
Department of Engineering- and Business Informatics
Kopernikusgasse 24/III, 8010 Graz, Austria
E-mail: dietmar.neubacher@tugraz.at

Michael O'Sullivan and Cameron Walker
University of Auckland
Department of Engineering Science
70 Symonds Street, 1010 Auckland
New Zealand
E-mail: michael.osullivan@auckland.ac.nz.

KEYWORDS
Hybrid Simulation, Model Design, Combined Simulation, Discrete Event Simulation, Agent-based Simulation

ABSTRACT

Simulation modeling is a powerful tool to help decision makers analyze complex systems. Existing techniques show great potential to capture dynamics of systems in some areas of application, but also have significant shortcomings in others. In this paper we present a hybrid approach which combines the computational advantages of Discrete Event Simulation (DES) with the flexibility of autonomous decision making and entity interaction provided by Agent-based Modeling and Simulation (ABMS). Therefore, we introduce a hierarchical model control structure that includes intelligence at the system level as well as at the individual level of entities. This allows the seamless inclusion of both non-intelligent and self-determined entities in a single model. Further, a combined time-advancement scheme is proposed that benefits from the advantages of the DES paradigm without losing the flexibility of ABMS. An example is presented that illustrates significant savings in computation time attainable by our hybrid approach, compared to using pure ABMS.

INTRODUCTION

Standard simulation approaches, such as ABMS, DES and System Dynamics (SD), all have distinct strengths and suitable areas of application. However, complex systems often consist of various different elements with diverse patterns of behavior. In many cases the nature of this behavior differs significantly over different parts of the model and hence is represented most efficiently by varying simulation paradigms. Thus, combinations of multiple paradigms for modeling and simulation of complex and large scale systems have gained significant attention over the last few years. (Brailsford et al. 2010; 2013, Djanatliev and German 2013, Viana et al. 2012). In this paper we focus on the hybridization of DES and ABMS models to benefit from the advantages of both methodologies. While ABMS provides the ability to describe individual behavior and interaction, DES is known to be more efficient in terms of computational performance. Although available computational power is constantly increasing, large scale problems still challenge modelers to develop the most efficient system representation possible. This is especially true when simulation is used as an evaluation tool for optimization algorithms, requiring multiple runs. For models consisting of both "intelligent" or self-determined agents and "non-intelligent" entities, efficiency can be gained by combining scheduled DES state-changes for entities and periodic updates for agents rather than solely using the ABMS approach.

Dubiel and Tsimhoni (2005) introduce a model of agent movement around a theme park. The individual movement of human beings, customers and staff, is modeled using agents. Trains run around the park on a tight schedule with distinct events changing their state, e.g. arrival and departure at stations, and are therefore represented as DES entities. Djanatliev and German (2013) combine a SD population model with an ABMS representation of health care facilities. Chahal and Eldabi (2008) classify modes of combining DES and SD. Brailsford et al. (2010) discuss challenges and state of the art with regards to the hybridization of DES and SD. An introduction and discussion of all possible combinations of paradigms, ABMS, SD and DES, is given by Heath et al. (2011). Huanhuan et al. (2013) introduce a framework to design hybrid ABMS/DES simulation on a more technical level than is presented in this paper.

Brailsford et al. (2013) and Viana et al. (2012) propose an integrated model of health and social care for the treatment of age-related macular degeneration that combines DES with SD. More importantly, they argue that the methodology behind most published hybrid models is determined by independent software packages linked via interfaces, or by software providing hybrid technology, e.g. AnyLogic and Repast. From a scientific point of view this is not very satisfying, because hybrid modeling is still in its infancy and should benefit from the freedom to explore new ideas, approaches and concepts. Being restricted by the features of commercial software has the potential to compromise the directions in which scientific endeavor can progress. Hence, we propose a methodology to model and implement truly integrated hybrid models, even if that results in losing the ease of drag and drop model generation provided by commercial software tools.

DISCRETE EVENT AND INDIVIDUAL BASED SIMULATION PARADIGMS

A vast number of publications deal with definitions, concepts and descriptions of ABMS (North and Macal (2007), Macal and North (2008)) and DES (Zeigler et al. (2000)) paradigms. In the following we present the basic definitions of both paradigms used in literature and discuss relevant features and fields of application.

Discrete Event Simulation

Discrete Event Simulation (DES) is a very powerful technique to simulate operational processes of entities that do not have the ability to make their own decisions. This is, in many cases, sufficient to capture a system's behavior. Its main advantage over ABMS is a more efficient time-advancement procedure that leads to higher-performance models. The most popular fields of applications for DES are the health care sector, manufacturing systems and the military sector, among others.

In its simplest representation DES is an iteration of the following steps: move to next scheduled event; check system; conditionally change system and schedule future events. How the system and its conditional fragments are represented and organized depends on the world view being used: Event Scheduling, Activity Scanning, the Three-Phase-Approach or Process Interaction, see Zeigler et al. (2000). Furian et al. (2014) presented a control based world view, Hierarchical Control Conceptual Modeling (HCCM) that is especially designed to describe systems with high entity interaction, as well as complex dispatching and control policies. HCCM can also be used as an implementation paradigm that combines features of the Three-Phase-Approach and Process Interaction world views. Furian et al. (2014) identify two types of conditional behavior (events and activities): requested behavior and system behavior. Requested behavior is motivated by an entity's own behavioral path (process) through the model. In other words, entities request certain activities and events dependent only on their own state and history (e.g. requesting a treatment). System behavior, on the other hand, is motivated by the system's state (including existing requests) and/or other control policies (e.g. moving to a patient for a requested treatment or workload/location balancing). The key element of HCCM models is control units. Control units manage entities, handle requests and trigger conditional behavior. Therefore, they include rule sets that determine which requests can be dispatched, how they are dispatched, what conditional (requested or system) behavior should be triggered and what requests are outdated and do not need to be kept any more. In other words they combine and centralize the conditions of activities and scanning of conditional process activations. Control units themselves are organized in a hierarchical tree form of models and sub-models.

The HCCM representation of DES is especially suited for hybrid ABMS/DES models, as the control units are very similar to the rule-sets associated with agents. Further, by generalizing world views, the HCCM representation removes the relevance of these world views (Heath et al. (2011)) when developing hybrid models, enabling the focus to remain on devloping the model rather than how the model should "fit" one of the world views.

Agent-based Modeling and Simulation

By using an agent-based or individual-based approach the system is modeled with autonomous self-organized entities that are called agents. Agents perceive their environment, execute various behaviors depending on their individual knowledge and interact with each other and/or the environment. The three elements of an agent-based simulation model are the agents themselves, their relationships, and the environment in which they are situated and with which they interact. Regarding their definition, agents should be self-contained and uniquely identifiable, autonomous and self-directed, social due to interactions with other agents and represented by states that vary over time. The flexibility of ABMS is its main advantage that and can accessed in many dimensions (Bonabeau 2002). To adjust the number of observed participants, agents can be easily added to/removed from the model. Furthermore, the complexity of an agent can be varied in many degrees. For example, the ability to learn or change existing behavior rules can be added. Finally, the ability to change the level of aggregation allows the modeler to use single agents and aggregated groups within the same model.

Due to their macroscopic view, passive entities and other limitations, techniques like DES and SD are not able to capture individual behavior and social interaction. Bonabeau (2002) stated the need for ABMS especially when there is potential for emergent phenomena, and this is most likely when individual behavior is nonlinear, exhibits memory or path-dependence or when interactions are heterogeneous and cause localized or network effects. Furthermore, real-world organizations consist of many different parts but do not necessarily call for a centralized controller. Hence, self-organization is common within almost any social system, because there is no central element that can control the behavior of each participant. However, there can be more influential elements such as dispatchers or classical behavioral codes.

Bonabeau (2002) classified the four most suitable areas of application for ABMS. First various kinds of human flows: for example road traffic, evacuation scenarios or passenger movement in airports or train stations. The second area is related to markets and strategic simulation where heterogeneous decision making plays a major role. The third class considers all kinds of organizations, especially its design and associated operational risk. The last area is called diffusion and is related to the diffusion of technology or innovation among customers and the underlying adoption dynamic. However, the benefit of an increased level of detail, which is supported by the ABMS approach, is directly linked to its big disadvantage - extensive computational effort.

HYBRID MODELS

The challenge of hybrid modeling is to elegantly combine features and elements of different paradigms in such way that the advantages of all methodologies can be utilized. At best this leads to higher simulation performance without the loss of modeling flexibility provided by either paradigm.

The combination of ABMS and DES elements is a natural approach, as they are similar in many aspects. Both consist of objects, either agents or entities, that change their state over time and engage in interactions. As Heath et al. (2011) pointed out, the main conceptual difference between the methodologies is that entities are simple data-containers with no individual decision making associated, where agents include rule-sets that represent behavioral strategies, cognitive aspects and communication with other agents. In other words, behavior of entities is strictly guided by a rule-set located at a system level, whereas agents act not only with respect to system rules and states, but also exhibit individual behavior. Obviously, ABMS provides more modeling flexibility. However, as agents require periodic state-updates, performance suffers compared to the event based scheduled time-advancement of the DES paradigm. In the remainder of the section we discuss different ways of combining ABMS and DES elements, revisit the concept introduced by Heath et al. (2011) and give the definition of hybridization proposed in this paper.

Types of Hybridization

Chahal and Eldabi (2008) classify concepts to hybridize DES and SD models in three different sets: hierarchical; process environmental; and integrated. During the discussion of all hybrid combinations of DES/ABMS/SD models Heath et al. (2011) made similar classifications with a slightly different nomenclature and dependent on the underlying DES world view. From summarizing both results it can be concluded that there are three fundamental ways to combine the above simulation paradigms, as shown by table 1.

Type	Description
Hierarchical	Models of different paradigms are simulated independently and results serve as input for other models
Parallel	All models are run parallel and exchange state variables
Integrated	Elements and concepts of all paradigms are integrated in a single model

Table 1: Types of Hybridization

Hierarchical approaches include designs of entirely separate and independent models that produce input measures for each other over iterative simulation runs. The benefit of such a design is that the models communicate only via defined interfaces. Apart from that they act entirely independently, which enables the use of almost all common software solutions, as long as they provide appropriate interfaces.

Parallel hybridization makes use of at least two paradigms for representation of different areas within a single model. However, these sub-models only interact by altering joint state variables. A classical example of this type is to represent environmental or aggregate states as a SD or ABMS model that serve as a parameter-set for a DES sub-model.

The far more challenging approach is to design integrated hybrid models via a seamless affiliation of paradigms. For ABMS/DES models this includes the interaction of entities and agents in the same environment. Modeling and description techniques of either methodology cannot be used in a straight forward way, as each lacks the tools to describe the elements and behavior of the other paradigm. According to Heath et al. (2011) the underlying DES world view has a significant influence on modeling integrated hybrid ABMS/DES systems.

Hybrid Simulation Software Tools

There are a few simulation tools which support hybrid simulation model generation. Primarily, Anylogic has attracted significant attention over the last few years, as it allows modelers to combine SD, DES and ABMS paradigms within a single model. The evolution of Anylogic is constantly progressing and it already supports the integration of agents as entities and resources within DES models. Hence, Anylogic is the only commercial software package taking steps towards integrated models so far. ABMS tools like Repast, Swarm or Netlogo provide interfaces where parallel or hierarchical combinations with other paradigms are possible, but these applications are still quite rare. Most commercial DES software packages, like Flexsim, equip entities with some sort of intelligence, e.g. choosing tasks from pools, that can arguably be seen as hybrid elements.

However, theoretical concepts on truly integrated ABMS/DES methodologies are extremely rare in literature. Hence, in this paper we propose an approach to model and simulate hybrid "individual-based" models, where entities and agents co-exist and directly interact in a single environment. In addition to the ideas from Heath et al. (2011), we abstract entities and agents to more general simulation objects with either non-intelligent, self-determined or dynamically changing behavior. Furthermore, we introduce a simulation and time-management procedure that combines the efficient event-scheduling methodology with periodic agent updates where required.

Hybrid Modeling using LEGOS

Heath et al. (2011) introduce a hybrid approach for combining ABMS and DES based on Listener Event Graph Objects (LEGOs) (Buss and Sanchez 2002, Schruben 1983). LEGOs are small encapsulated DES components that interact via an event-source-listener relationship. In particular, if an event A occurs in a source LEGO S, all LEGOs listening to S that contain an event with the exact same name, A, trigger their version of that event.

Heath et al. (2011) argue that agents can be represented by single LEGOs. State changes are modeled via periodically recurring events. Further, rule sets are incorporated by conditions and state changes associated with events. Hence, agents can be seamlessly integrated in the DES environment, as they are basically DES submodels themselves.

Although, this is a very general and fundamental definition of hybrid ABMS/DES models, it is the belief of the authors that the ease of modeling can be increased by the explicit definition of simulation objects (entities or agents in the common sense) and the inclusion of additional forms for the representation of interfaces and rule sets. The use of activities and events as the main modeling tools is also encouraged by the methodology proposed in this paper

Definition of Integrated Hybrid ABMS/DES Models and Naming Conventions

In the combination of ABMS and DES the observed system elements should be named objects rather than entities or agents. Such objects can be either non-intelligent or intelligent in any kind of form. Furthermore, they are able to change their level of abstraction over time. As a result non-intelligent objects, acting like discrete event entities, and self-organized agents coexist and interact within a single model. In particular, they engage together in activities and events that can affect the state of both of them. Further, the behavioral path of classical entities (non-intelligent) is affected by the behavior of classical agents (intelligent) and vice-versa.
Hence, common naming conventions for elements of both DES and ABMS have to be revisited and integrated for hybrid models, as illustrated by table 2.

	DES	Hybrid	ABMS
Elements	Entities	Objects	Agents
(Inter-) Actions	Activities/ Events	Individual CU/ Messaging & Events	Behavior
System	Rule Sets/ Control Unit (CU)	System CU/ Interfaces/ Interface Controller	Topology

Table 2: Hybrid Naming Conventions

Simulation Objects

A simulation object may consist of up to four elements, as illustrated by figure 1: Interface (I); the corresponding Interface Controller (IC); an optional Individual Control Unit (ICU); and a data block (containing attributes and state variables).

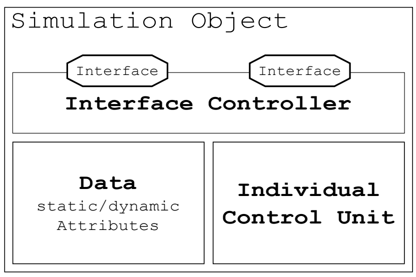

Figure 1: A system object contains four different parts

As mentioned above, hybrid models include intelligent and non-intelligent objects at the same time. Hence, non-intelligent, or system-directed, objects do not include an ICU. It is common practice for ABMS modeling to consider two levels of rule sets: base-level rules and higher-level rules. Base-level rules define the response and interaction with the environment, while higher-level rules let the agent's behavior adapt to it's current state (Casti 1997). In terms of hybrid modeling both are included in the ICU and allow an almost continuous variation of objects' intelligence. Furthermore there is the possibility to switch the objects ICU entirely off, which is situated in the IC. Consider the example of people moving around in trains. Their behavior during train rides might not be of interest at all, hence, their ICU is switched on and off upon *enter* and *leave* events. Note that these changes not only affect the objects behavior, but also may have an influence on time advancement. Different states, or activities, of objects and associated base-rules may require varying time-intervals for periodic updates.

Model Structure

The benefit of structuring models in a hierarchical model-sub-model structure to organize larger systems has been identified by Zeigler (1987). Therefore, this hybrid approach tries to build sub-models similar to the methodology proposed by Furian et al. (2014) for DES (see figure 2). The encapsulated system, or environment which is identified as a proper sub-model also consists of three elements: interface (I); System Control Unit (SCU); and data block. In contrast to individual objects the SCU includes all intelligence nested at this sub-system level, such as rules for dispatching or reactions to the system's state. Furthermore, it defines the environment of the sub-model and manages changes to the latter. Objects and the system communicate in a bidirectional way via requests, messages and event listening, as will be described in more detail below.

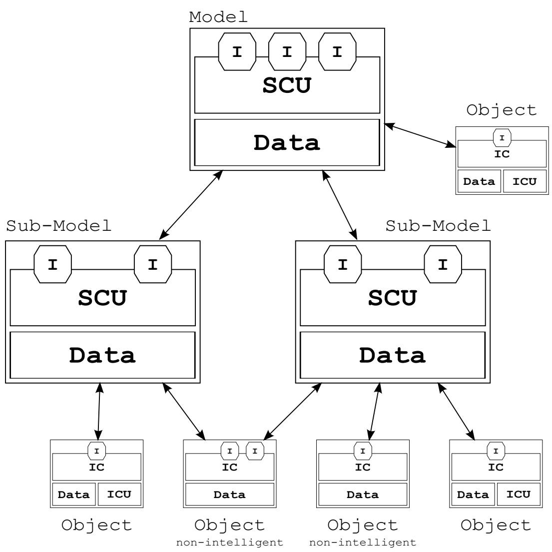

Figure 2: The hierarchical structure using sub-models (containing the SCU) provides a more realistic model structure

Events and Activities

Heath et al. (2011) argue that events and activities are the most natural way to describe the behavior of many platforms. In particular, humans engage in a certain activity until finished or interrupted and then proceed with the next one.
Events happen at discrete points in time and change the state of participating objects and/or the system in corresponding state-changing routines. Activities stretch over a non-zero amount of time and are bounded by a start and end event. For the hybrid methodology presented in this paper they play a key role, as they conceptually and technically link intelligent and non-intelligent objects. In particular, both object types engage in common activities that change their state and, in the case of intelligent objects, also affect their behavior during participation. Recall that there are rules for adaption and "rule-changing rules". It seems natural that these higher-level rules are dependent on the activity an object is currently engaged in.
How events and activities are triggered in DES models is dependent on the world view utilized. However, according to the concept described by Furian et al. (2014) events and activities are triggered either at a system level with respect to the evaluation of rule sets, in a classical scheduled DES way, or sequentially from other events. Following that logic, in our hybrid modeling, events can be triggered by both types of control units (SCU and ICU), in a scheduled fashion (e.g. periodic update events) and sequentially from previous events.

Object Interaction

One of the main features of ABMS is agent interaction. Furian et al. (2014) introduced the concept of requests for DES, that communicate an entity's desire to participate in a certain activity, which may be also seen as a form of interaction between entities and the system.
In this paper we propose three fundamental forms: event listening; requests; and customized interfaces.
Event listening, as proposed by Buss and Sanchez (2002), allows elements of the model to react to triggered events. As the reaction requires some sort of intelligence, it is only applicable for objects containing an ICU or the system level and its SCU. Non-intelligent objects may also be affected by triggered events, as their state might get changed or an ICU will be activated, but they do not actively listen for them. As an example imagine a doctor that is assigned to treat a patient. He may immediately treat the patient, or engage in another activity first, in contrast to DES where the control unit instantly triggers the start event of the treatment activity as soon as it is dispatched and resources are available. Furthermore, objects may not only file requests to the SCU, but also directly to other objects. To be able to deal with this more complicated behavior we allow any

kind of control unit, SCU and ICU, to handle and process requests. Furthermore, all elements, objects and the system itself are able to file requests.

Requests, on the other hand are data-holding elements that are passed around the system. For DES they are filed by entities according to their behavioral path, or process, and are handled by a control unit. They not only act as queuing mechanisms, but provide more flexibility for dispatching. For hybrid models it seems beneficial to allow objects to file requests for activities or events at the system level. Consider patients in a health care facility that require some sort of treatment. Upon arrival, or completion of a previous activity, they file a request for the desired action which is then held and processed by other elements of the system. As pure DES only contains non-intelligent objects that are not able to autonomously trigger behavior, these requests are handled only by control units on a system level. However, as objects in hybrid ABMS/DES also include self-determined behavior, handling of requests becomes more complicated. To be able to deal with this more complicated behavior we allow any kind of control unit, SCU and ICU, to handle and process requests. Furthermore, all elements, objects and the system itself are able to file requests. In other words, the system is able to file a request to a intelligent simulation object, e.g. telling the doctor and the patient to engage in the treatment activity.

As there is always the possibility that complex systems contain some sort of messaging, perception or interaction that may not be elegantly modeled using requests or event listening, we also consider custom interfaces between any kind of control units.

In terms of implementation, interaction is handled the following way. By the time a control unit (ICU or SCU) triggers an event, the corresponding IC checks if a channel for the event is included in the associated interfaces. If so, the event is processed through this channel to it's listening IC. The listening IC passes the information to it's control unit. Note that it does not make sense for non-intelligent objects to listen for events, as they do not have the ability to process them. In this case the event itself would already change the state of the object. Requests on the other hand, are either filed by a control unit or an IC of non-intelligent objects. When filed by control units they represent dispatching or policy decisions that require objects to engage in certain activities, e.g. treat that patient, or the need of an object to engage in an action, e.g. a patient requiring a treatment. However, non-intelligent objects (without an ICU, or while it is switched of) are also able to file requests according to their behavioral path through the model. Upon the completion of an activity the following action is looked up in the object's process and is requested via the object's IC. It seems beneficial for implementation if the information of an object's process is stored in the object itself. Hence, the IC should provide a methodology to request the next activity which is then called in the state changing routine of the finishing event.

TIME ADVANCEMENT

In the last section we proposed a methodology to hybridize ABMS and DES in a conceptual way. This section introduces concepts for a joint time advancement mechanism. We briefly outline common time advancement techniques for DES and ABMS before presenting a combination of both.

DES and ABMS Simulation Sequences

DES simulation progress is simple and fast. Figure 3 shows the basic iterative principle of the DES simulation sequence: At each event evaluate the system (1); check conditional behavior (2); change system's state; if necessary schedule future events (3); If no new behavior, advance to next scheduled event (4). Hence, time is advanced only at discrete points in time where events happen, with no state-checks and updates performed in between.

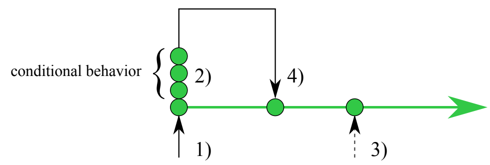

Figure 3: Time Advancement in DES

Agent-based simulation normally proceeds time in small "user-defined" time steps. Each time step all agents' states are updated, with respect to their rule sets and communication interfaces. A slightly different approach was proposed by Heath et al. (2011), where agents are small encapsulated discrete event models that move from event to event and thereby change their state. The big advantage of this idea is that intervals between updating events are not necessarily defined by a fixed constant and it gives the opportunity to schedule events between (most likely existing) periodic updates. Buss and Sanchez (2002) even proposed a movement-model based on this methodology.

Hybrid ABMS/DES Time Advancement

We utilize the ideas of Buss and Sanchez (2002) and Heath et al. (2011) for the hybrid approach proposed in this paper. In particular, we assume that the state of intelligent objects is changed in a series of events. This series includes periodically reoccurring events, as well as non-periodic events in-between.

Each event may trigger the evaluation of the agent's rule set hosted in its ICU, which possibly leads to triggering of additional behavior and/or filing requests. Extraordinary events can occur for various reasons, including all communication schemes and object-internal procedures. Allowing non-periodic object updates enables investigation of less periodic state changes without losing too much accuracy in system representation. Furthermore, the rate at which periodic events occur may be dependent on many factors, including the activity the object is currently engaged in and/or the system's state. Some tasks that cannot be interrupted and are not in the focus of the simulation may not be modeled in an ABMS context at all, e.g. sleeping. In this case the object does not update periodically at all and it's state remains constant until the next event is triggered at the system level or in a scheduled fashion.

The main goal of a hybridized approach is to combine both time advancement procedures in a single time line. Thereby, non-intelligent objects are only updated upon non-periodic events, which significantly enhances performance. Further, the evaluation of intelligent objects' rules is performed upon periodic and non-periodic events. This may lead to instance changes of the object's state, as well as triggering of sequential events that may also affect other objects.

All possible future events are held in a global list that is sorted with respect to scheduled times, including periodic update events. Hence, the simulation clock proceeds to the next scheduled event, as in DES, triggers all events scheduled at that time and iteratively performs rule sets of all SCUs in a top-down manner according to their hierarchical structure until no further action (firing of events and/or interaction) has been observed.

In terms of implementation it is strongly recommended that periodic update events of objects are aggregated into one single event per time point that triggers the evaluation of agents' rule sets via a provided interface (e.g. event listening). Furthermore, update events should be scheduled at definite points in time, e.g. objects are updated every full second to avoid objects with the same periodic interval being updated at arbitrary time-points.

Note that software packages like Anylogic also include a combined mechanism that allows agents to communicate and evaluate their state at scheduled events and discrete time steps. However, the approach presented above is more flexible as discrete time steps are replaced by periodic scheduled events and a corresponding listener pattern. This has two major advantages: first, it is not necessary to consider the entire set of simulation objects for periodic updates; and second, objects can dynamically adapt the frequency of rule-evaluation with respect to their current state and are not bound by a rigid interval.

To sum up, periodically and non-periodically scheduled, conditional and sequential events may occur in the system. Sequential events are triggered in the state-changing routine of previous events. Conditional events are triggered by any kind of control unit, SCU or ICU, during rule evaluation and scheduled events happen at distinct points in time. Events can be scheduled during the state-changing routine of previous events, e.g. within the start event of an activity its end event is scheduled, or by rule sets themselves.

An example of a combined time line is given in figure 4. At time point 1 the system is initialized, which leads to updates of DES and ABMS like components. Next, two intelligent objects are updated upon the arrival of a periodic event. Time point 3 represents a scheduled event that inter-periodically changes the state of object 2. Further, we indicate that certain events may conceptually bound activities that objects are engaged in.

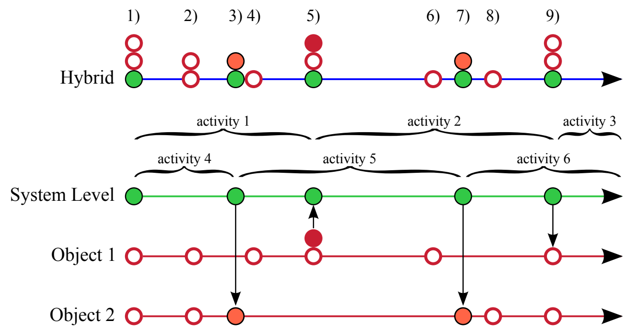

Figure 4: Time advancement of the hybrid approach

WORKING EXAMPLE

To illustrate the benefits of the proposed approach we discuss a problem introduced in an ABMS tutorial by Macal and North (2008). We design a classical ABMS model, as well as two hybrid versions that follow the methodology presented in this paper. Due to the simplicity of the example, its purpose is not to illustrate all introduced features and concepts, but to demonstrate that more advanced modeling paradigms do lead to better performing simulations.

All models were implemented from scratch in C# to compare resulting run-times and model efficiency.

Problem Description

The problem is defined as follows. The environment consists of a $n \times n$ grid of quadratic cells, each of the same size x. Cells have two states, green (active) and red (inactive). They change their state at random points in time where the duration between changes is uniformly distributed over intervals $[l_g, u_g]$ for green and $[l_r, u_r]$ for red. Furthermore, m individuals live on that grid, which move on a green cell or remain at a constant position on a red cell. In particular, individuals moving on a cell turning red stop immediately or stop upon entering a red cell. Individuals move with constant speed v in an orthogonal direction ($\in \{0°, 90°, 180°, 270°\}$ with respect to the grid) that is randomly chosen upon the initialization of the problem. The start position of individuals is randomly chosen on the grid. As soon as they move off the grid they are not considered for further changes. The simulation terminates as soon as all agents have left the grid.

Note that this very simple example could be designed as a pure DES model. However, as soon as more complex behavior of individuals on green cells is included (as collision-avoidance or speed and direction changes) the features of ABMS simplify the modeling significantly.

Model Implementation

We implemented three simulation models of the problem: a pure ABMS implementation; a hybrid (H) approach; and a semi-hybrid approach in which only cells are updated in a DES way (HOC).

The ABMS implementation follows exactly the methodology proposed by Macal and North (2008). Time increments are given by a constant and arbitrary value Δt. Each time the clock advances, possible state-changes of all cells are considered. Subsequently, positions of all individuals are updated. Therefore, an individual assesses it's environment, e.g. the color of the cell it's currently on, and moves with speed v in his designated direction if located on a green cell. To reduce the number of method calls that determine the hosting cell of a specific position, each object stores the identity of the current cell it is on.

Within the H approach cells are engaged in cell-changing events that invert the state of cells. These events are scheduled during initialization of the model and by their previous instances, e.g. an event changing cell i schedules the next event for cell i and so on. Furthermore, cell-changing events change (or trigger) individuals' current activities on the cell, which defines the degree of intelligence associated with the individual. During waiting any self-determination is switched off and no periodic update events are required. Periodic update events are scheduled every Δt time steps by their previous event (and upon initialization). Moving and remembering of the current cell is modeled in the same way as in the ABMS approach. Note that, in this example, object interaction is rather limited. No requests are filed, individuals listen to update events by evaluating their rule sets hosted in ICUs and are directly affected by cell-changing events.

The HOC implementation updates only cells in a scheduled fashion. Individuals on the other hand are constantly checked for moving. This allows the analysis of the contribution of both features (scheduled cell updates and partly scheduled individual updates) to total runtime savings.

Experimental Results

To analyze performance of all implemented approaches several instances have been generated with various combinations of: intervals for the uniform distribution of green and red phase durations; and different numbers of individuals. The initial states have been generated with probabilities $p_g = \frac{u_g}{u_g+u_r}$ for green and $p_r = \frac{u_r}{u_g+u_r}$ for red respectively. Note that the very first phases of green were chosen uniformly distributed over intervals $[0, u_g]$ and $[0, u_r]$ to increase the variability of cell-changing times among different cells. Further selected parameters are $n = 12$, $x = 25$, $\Delta t = 1$, $m \in \{100, 1000\}$ and individuals move at speed 1 per time step.
For each parameter set 50 instances were simulated and mean run-times and the exit times of the last individual were reported. Runtimes of each hybrid approach were divided by run-times of the ABMS simulation to obtain relative savings. Results are summarized by figure 5.

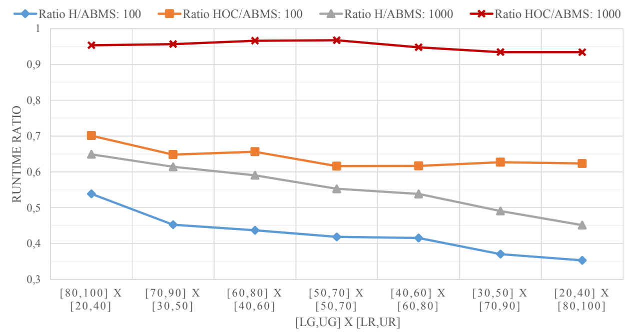

Figure 5: Runtime Analysis

As expected we can observe that run-time savings for the full hybrid approach increase (decreasing ratio) for longer red-phases, as the ABMS simulation performs more update-calls. Note that duration of green and red phases has no significant influence on performance of the HOC simulation and almost no performance enhancement for instances with 1000 individuals can be observed using this method. This is an expected behavior since cell updates consume a fixed absolute amount of time (independent of the number of individuals), which contributes less to relative savings for increasing total runtimes.The absolute savings due to the H approach are due to the savings from: 1) using DES for the cells, which is a fixed saving independent of the number of individuals; and 2) using DES for the individuals which is a saving that will increase as the number of individuals increases. In our example, the relative savings from the cells is large when compared to the relative saving for each individual. Thus, as the number of individuals increases and the savings from the cells becomes negligible, the relative savings will decrease to match that from the individuals, hence the decrease in relative savings for 100 individuals when compared to 100 individuals.

CONCLUSION AND FUTURE WORK

In this paper we argue that by combining DES and ABMS simulation paradigms the modeling flexibility of ABMS can be preserved while utilizing the run-time efficiency of the DES methodology. Further, general design concepts for hybrid DES/ABMS models that allow a seamless combination of both approaches are introduced. Thereby, model layout structure, object definitions and interactions as well as simulation control on a system and individual level are outlined.

The new methodology is tested on a basic ABMS model. For these experiments the hybrid approach is more efficient than the pure ABMS simulation. Even for such a simple example a significant gain in performance can be achieved by designing models that combine the strengths of the ABMS and DES paradigms.

In ongoing simulation studies the authors will apply the introduced concepts to more complex real-world applications, mainly in the health care sector.

ACKNOWLEDGEMENTS

The first author thanks the Austrian Science Fund (FWF): Project Nr. J3376-G11, for funding his research in New Zealand.

REFERENCES

Bonabeau E., 2002. *Agent-based modeling: methods and techniques for simulating human systems.* In *Proceedings of National Academy of Sciences*. vol. 99(3), 7280–7287.

Brailsford S.C.; Desai S.M.; and Viana J., 2010. *Towards the holy grail: Combining system dynamics and discrete-event simulation in healthcare.* In *Proceedings of the 2010 Winter Simulation Conference*. 2293–2303.

Brailsford S.C.; Viana J.; Rossiter S.; Channon A.; and Lotery A.J., 2013. *Hybrid simulation for health and social care: The way forward, or more trouble than it's worth?* In *Proceedings of the 2013 Winter Simulation Conference*. 258–269.

Buss A. and Sanchez P., 2002. *Building complex models with LEGOs (Listener Event Graph Objects).* In *Proceedings of the 2002 Winter Simulation Conference*. vol. 1, 732–737.

Casti J., 1997. *Would-Be Worlds: How Simulation is Changing the Frontiers of Science.* Wiley.

Chahal K. and Eldabi T., 2008. *Applicability of hybrid simulation to different modes of governance in UK healthcare.* In *Proceedings of the 2008 Winter Simulation Conference*. Miami, FL, 1469–1477.

Djanatliev A. and German R., 2013. *Prospective healthcare decision-making by combined system dynamics, discrete-event and agent-based simulation.* In *Proceedings of the 2013 Winter Simulation Conference*. 270–281.

Dubiel B. and Tsimhoni O., 2005. *Integrating agent based modeling into a discrete event simulation.* In *Proceedings of the 2005 Winter Simulation Conference*. 1029–1037.

Furian N.; Vössner S.; OSullivan M.; and Walker C., 2014. *HCCM-A Control World View For Health Care Discrete Event Simulation.* In *Proceedings 28th European Conference on Modelling and Simulation.*

Heath S.K.; Brailsford S.C.; Buss A.; and Macal C.M., 2011. *Cross-paradigm simulation modeling: Challenges and successes.* In *Proceedings of the 2011 Winter Simulation Conference*. 2783–2797.

Huanhuan W.; Yuelin Z.; and Meilin Z., 2013. *A framework for integrating discrete event simulation with agent-based modeling.* In *Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on*. vol. 3, 176–180.

Macal C.M. and North M.J., 2008. *Agent-based modeling and simulation: ABMS examples.* In *Proceedings of the 2008 Winter Simulation Conference*. 101–112.

North M.J. and Macal C.M., 2007. *Managing Business Complexity : Discovering Strategic Solutions with Agent-Based Modeling and Simulation: Discovering Strategic Solutions with Agent-Based Modeling and Simulation.* Oxford University Press, USA.

Schruben L., 1983. *Simulation Modeling with Event Graphs. Commun ACM*, 26, no. 11, 957–963.

Viana J.; Rossiter S.; Channon A.A.; Brailsford S.C.; and Lotery A., 2012. *A Multi-paradigm, Whole System View of Health and Social Care for Age-related Macular Degeneration.* In *Proceedings of the 2012 Winter Simulation Conference*. WSC, WSC '12, 95:1–95:12.

Zeigler B.; Praehofer H.; and Kim T., 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press.

Zeigler B.P., 1987. *Hierarchical, modular discrete-event modelling in an object-oriented environment. SIMULATION*, 49, no. 5, 219–230.