



Multi-Perspective Modeling of Healthcare Systems

Ignace Djitog, Hamzat Olanrewaju Aliyu, Mamadou Kaba Traoré

► To cite this version:

Ignace Djitog, Hamzat Olanrewaju Aliyu, Mamadou Kaba Traoré. Multi-Perspective Modeling of Healthcare Systems. International Journal of Privacy and Health Information Management, 2017, 5 (2), pp.1 - 20. 10.4018/IJPHIM.2017070101 . hal-01814162

HAL Id: hal-01814162

<https://hal.science/hal-01814162>

Submitted on 20 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Perspective Modeling of Healthcare Systems

Ignace Djitog

*African University of Science and Technology
Abuja, Nigeria
djitog2@gmail.com*

Hamzat Olanrewaju Aliyu

*Federal University of Technology
Minna, Nigeria
hamzat.aliyu@futminna.edu.ng*

Mamadou Kaba Traoré

*Université Blaise Pascal
Clermont-Ferrand, France
traore@isima.fr*

ABSTRACT

This paper presents a multi-perspective approach to Modeling and Simulation (M&S) of Healthcare Systems (HS) such that different perspectives are defined and integrated together. The interactions between the isolated perspectives are done through dynamic update of models output-to-parameter integration during concurrent simulations. Most often, simulation-based studies of HS in the literature focus on specific problem like allocation of resources, disease propagation, and population dynamics that are studied with constant parameters from their respective experimental frames throughout the simulation. The proposed idea provides a closer representation of the real situation and helps to capture the interactions between seemingly independent concerns - and the effects of such interactions - in simulation results. The article provides a DEVS (Discrete Event System Specification)-based formalization of the loose integration of the different perspectives, an Object-Oriented framework for its realization and a case study as illustration and proof of concept.

Keywords: Healthcare Systems, Multi-perspective M&S, Parameterized DEVS, Output-to-parameter Integration.

INTRODUCTION

Being composed of concurrent, fragmented and diverse components interrelated with intricate processes, modeling the domain of healthcare will require the understanding of the behavior of the overall system (Barjis, 2011). Decision-making concerning questions related to the performance of HS - such as the extent to which the system achieves its mission - have no clear or simple answers while the need to produce more with less resources despite the scarcity is becoming a widely acknowledged concern among policy-makers and healthcare managers worldwide (Shin et al. 2013). This is proven by a considerable volume of work published in recent years being dedicated to simulation-based study of HS. Frequently, modeling approaches used to investigate different aspects of HSs related to healthcare simulation include discrete event simulation, mixed method that combines simulation with optimization techniques (Ahmed and Alkhamis, 2009), goal programming (Topaloglu, 2006) and discrete event simulation with data envelopment analysis (Weng et al. 2011). Arguably, unit specific studies of simulation modeling in healthcare that deals with specific problems have been predominant in the published research articles. Such unit specifics include outpatient clinics, A&E (Accident and Emergency Departments), and inpatient facilities and are presented by Khurma et al. (2013), and Choi et al. (2013). The common issues addressed in the literature include, but not limited to scheduling and patient flow, sizing and planning of beds, rooms, and staff.

The main challenge in modeling a complex system such as HS is the large number of its components and their diversity. To the best of our knowledge there has not been a generic model of healthcare simulation that considers the different elements of HS and their interactions to describe a complete whole. This paper investigates HS through multi-perspective modeling and addresses the challenges that come with modeling such a complex system. Multi-perspective modeling allows constructing distinct and separate models from different aspects of HS for a better understanding of its complexity. Furthermore, an integrative approach based on live updates of output-to-parameters translation is developed to allow the simulation output of a model of a given perspective to update the simulation parameters of another perspective dynamically. Arguably, a closer representation of the real situations can be achieved if these parameters are systematically modified at runtime in such a way that the outputs of the simulation models corresponding to different perspectives provide live updates of their parameter(s) in concurrent simulations. Therefore, the formalization of the bases of the proposal is provided with a case study that presents the models of different perspectives in HS and shows how their integration is being achieved.

The rest of this paper is organized as follows: the next section presents the literature review followed by the multi-perspective modeling of HS. The DEVS-Based formalism for integrating HS perspectives is then presented with a case study to illustrate its application before concluding the paper with directions for future work.

LITERATURE REVIEW

Modern HSs have been explored with a variety of studies over many decades. Although not exhaustive, a number of examples of these studies include discrete event simulation, system dynamics, agent-based simulation, Monte Carlo simulation, hybrid simulation (combination of discrete-event and continuous) and simulation combined with optimization techniques. Roberts (2011) presented an extensive tutorial of such simulation modeling methods with a revision of taxonomy of the use of computer simulation in healthcare into two categories: Patient flow optimization and Analysis, and healthcare asset allocation. More specifically, Gunal and Pidd (2010) enlarged this taxonomy in a review of the literature for discrete event simulation for performance modelling in healthcare into scheduling and patient flow, sizing and planning of beds, rooms, and staff. Unit specific studies of simulation modeling in healthcare focusing on solving specific problems in individual HS units such as Outpatient clinics, A&E (Accident and Emergency Department), and Inpatient facilities are predominant in healthcare simulation literature.

Consequently, most of the published articles look into HSs with focus on a single perspective modelling. For example, Ozcan et al (2011) examined patient pathway across different healthcare units to identify critical activities and scarce resources representing process bottlenecks. A related study by Cote (1999) examined the daily arrivals and the throughput of patients to analyze the utilization and the allocation of examining rooms at a family practice clinic. Other than patient flow and resources utilisation perspectives, is the simulation modelling related to demand that includes epidemiology research and health policy-making. For example, an integrated agent-oriented modelling and simulation framework that considers both population and healthcare delivery network has been introduced by Charfeddine and Montreuil (2010). Okhmatovskaia et al. (2012) presented ontology for simulation modelling of population health (SimPHO) a formal, explicit machine-readable specification of a domain of knowledge

integrating both aspects of taxonomy and vocabulary in a form of logical axioms. A perspective related to disease spread has also been considered by several models to predict its propagation in a pandemic and the effect of HSs intervention and is reported by the work of the authors Kasaie et al. (2013), Carr and Roberts (2010), and Dibble (2010).

Similarly, simulation has been used as main tool for training medical students through studies such as serious games. Bruzzone et al. (2012) proposed an advanced serious game called MARIA (Model for Advanced and Realistic patient simulation driven by Intelligent Agents), that aims at supporting and renewing educational processes in healthcare care sector. The developed tool helps teaching doctoral students to take care of their patients during the entire patient life cycle while overcoming their needs and fears. The virtual patients being simulated are driven by intelligent agents and characterized by realistic human behaviors such as pain, fear, and psychological issues. Bonnetain et al. (2010) presented a study on the crucial role of computer screen-based simulator, called MicroSim in training medical students in cardiac arrest procedures for acquiring basic skills during cardiopulmonary resuscitation (CPR) procedures. Another educational tool called SimMan, known as high-fidelity patient simulator was also introduced for the test session in the study.

Based on the information provided in the review of health simulation models, the level of details is often insufficient and remains a key issue to understand precisely how the elements of HS are interrelated and what relationships and interactions are defined between the underlying components. As a proof of concept, most of the reported studies are either focusing on the specific problems in individual units of HSs or are built for specific health facility (Gunal and Pidd, 2010). To address the issues outlined above, the authors present a multi-perspective modeling approach that describes how the different aspects of HSs can be modelled with specific focus at different levels of abstraction and be linked together to form an integrated whole.

Multi-perspective modelling has been successfully applied in other areas to study complex systems. According to Seck and Honig (2012), to understand complex systems one of the common ways is to impose on them a hierarchical design by separating the subcomponent parts and defining relations between them. Modelling and simulation approaches such as DEVS formalism (Zeigler et al. 2000) rooted in system hierarchy specification formalism, represents systems as atomic and coupled models. System of systems approach is another way used to study complex system. As such, in a systematic view of healthcare, Zeigler et al. (2012) presented an illustrative general framework for systems of system level modeling to support design of coordinated care architectures for linking up hospitals and physicians through networked information systems. Modeling complex system within a single model where different aspects of the system are captured by different views has been reported as difficult, if not impossible task by Reineke and Tripakis (2014). As such, Seck and Honig (2012) proposed a multi-perspective modelling approach to overcome the limitations of a single-perspective hierarchy approach based on aspect models integrated with bridge models. The authors adopted the idea of modeling relation as introduced by Rosen (2000), and Mikulecky (2001) that defines a bridge between two worlds as the natural world in which we live and the mental world referred to as formal system that represents our perceptions of the former. Based on this idea, they argued that when trying to model a complex system, it can be described as a collection of perspectives where each perspective represents a unique formal system having a unique decomposition. Hence, one could model a complex system in a richer way by having multiple non-isomorphic decompositions that may influence each other and capture its complexity resulting from the observation through different perspectives. Tekinay et al. (2010) introduce a context-based

“view” concept as an enabler to support multi-perspective modeling in multi-actor environments using an example of quay crane system. Also, Braun and Esswein (2015) argued that structuring the different views of an information system through multi-perspective modeling is more relevant to improve the understanding of its complexity. Multi-perspective modeling has also been used by Kingston (2001) in ontology to represent knowledge from the viewpoints of “who”, “what”, “where” and “how” for the purpose of knowledge valuation. The argument is based on a widely acknowledge concern which states that the principle of multi-perspective modeling is that for any “knowledge asset” to be represented adequately, it’s necessary to represent a number of different perspectives on its knowledge and, possibly, to represent the asset at multiple different levels of decomposition. We highlight some key contributions of the paper:

- It offers to the modeller, through separation of concerns, a clear view on perspectives like patient flow optimization and analysis, healthcare resource allocation, and disease propagation control that are most often intertwined (Gunal and Pidd, 2010). In this paper, the different concerns are separated into four generic perspectives for a holistic view of HSs modelling.
- It proposes a multi-perspective modeling approach of HSs to overcome the problems of single-perspective modeling used in solving problems in individual HS units like outpatient clinics, A&E, and Inpatient facilities, as well as facility specific problems. The proposed approach presents a broader view on healthcare modelling including various perspectives than the ones proposed by Charfeddine and Montreuil (2010) that dealt specifically with population and healthcare delivery network perspectives.
- It presents a novel approach for integrating the isolated perspectives in HSs based on dynamic update of models output-to-parameter integration during concurrent simulation contrary to the classical models coupling through outputs and inputs interfacing of simulation models (Zeigler et al. 2000) with parameters that are usually assumed to be constant throughout the simulation.
- It provides a DEVS-Based formalization of the loose integration of the different perspectives, and an Object-Oriented framework for its realization. DEVS formalism is chosen because it is universal for discrete event simulation as proven by Vangheluwe (2000) while Object-Oriented design patterns are reusable solutions to some general software engineering problems. This solution differs from the one proposed earlier by Seck and Honig (2012) that represents the dynamic parameters as input ports that are coupled to the respective sources of live feedbacks. Furthermore, concrete implementations of both aspects are provided while to the best of the authors’ knowledge these aspects have not been addressed both together before.

The authors view HSs as a collection of perspectives that are each associated with a unique component system. A case study is presented to illustrate the concurrent simulation of the different perspectives through live update of simulation parameters.

MULTI-PERSPECTIVE MODELING OF HS

This section extends the work of Djitog et al. (2015) which proposed a disciplined stratification of the various concerns studied in HS simulation into four layers/perspectives: Allocation of Healthcare Resources (AHR), Health Phenomena Dynamics (HPD), Population Dynamics (PD)

and Individual Behaviour (IB). These perspectives are described by the positions from which one can look at the HSs as depicted in Figure 1 showing arrows that indicate the current positions of the viewer. Hence, modelling HS through multi-perspective modelling becomes more practical and richer with deep insights. In Figure 1, AHR addresses the problems of scarce human and infrastructural resources deployed to offer healthcare services. HPD encompasses problems like disease outbreaks, chronic ailments like cancer, hypertension, diabetes, antenatal program to follow up pregnant women, assistance to elderly people and some other important issues related to social care such as home care and appointment attendance monitoring. PD covers the influences of immigration, emigration, birth and death on community health while IB includes human behaviour-related issues like educational level, physical state, emotion, cognition and social status.

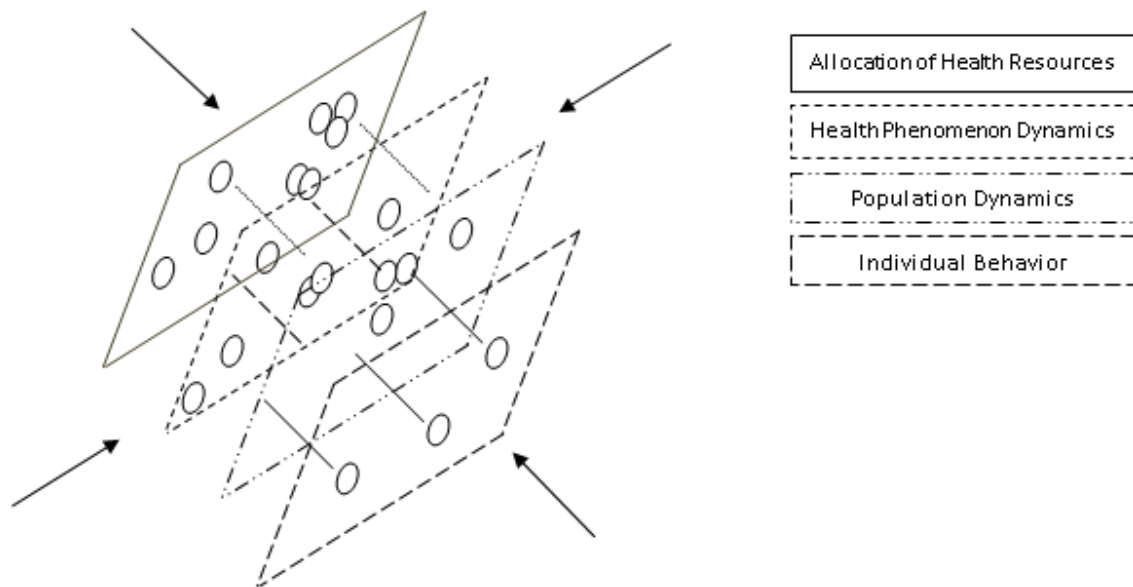


Figure 1. Perspectives on HSs.

Figure 2 illustrates the proposed integration of concurrent simulation processes of the different layers (perspectives) of HS. Considering that the models and experimental frames of layers A, B, C and D correspond to the simulation setup for AHR, HPD, PD and IB perspectives respectively. The parameters $p_1 - p_n$ in each layer represent certain outputs of the simulation processes in other layers. While these parameters are usually considered to be constant through a simulation run, the authors argue that, in reality, the original properties represented by any of the parameters can change at runtime thereby making the parameter value outdated in subsequent use. For instance, given an hypothetical simulation model M_{alloc} for AHR in a healthcare facility as a response to an epidemic in the immediate environment of the facility. Imagine there is another model M_p of the epidemic itself in the context of the same environment and one of the outputs, y , of M_p is the percentage of the infected population. Assuming M_{alloc} has a parameter, x , that denotes the estimated number of infected inhabitants; rather than keeping x constant throughout a simulation run, the idea proposed in this paper is to run the simulations of M_{alloc} and M_p concurrently in their

respective experimental frames such that, instantaneous outputs, y , of M_p are used to update the parameter, x , of M_{alloc} at runtime.

Therefore, this paper proposes an integration mechanism to connect relevant outputs of a model to the parameters representing them in other layers. This integration mechanism is illustrated by the dashed lines connecting some output ports to parameters in other layers. For instance, port o_1 of layer A is connected to parameter p_2 in layer B while o_3 is connected to p_n and p_2 of layers C and D respectively. The purpose of these output port-to-parameter connections is to enable live update of the parameters at runtime whenever there are changes in the values of the actual properties they represent. The authors believe this approach gives a better representation of the influences of the different perspectives of HS on one another and will produce simulation results that are closer to the real behaviours of the actual systems in this context. A DEVS-based formalism will be provided in the next section to guide the application of this idea to concurrent DEVS simulation process of the different perspectives. Of course, models of different perspectives may be best specified in different formalisms, the authors have chosen DEVS to present the idea in this paper because it is considerably universal for discrete event simulation as proven in (Vangheluwe 2000).

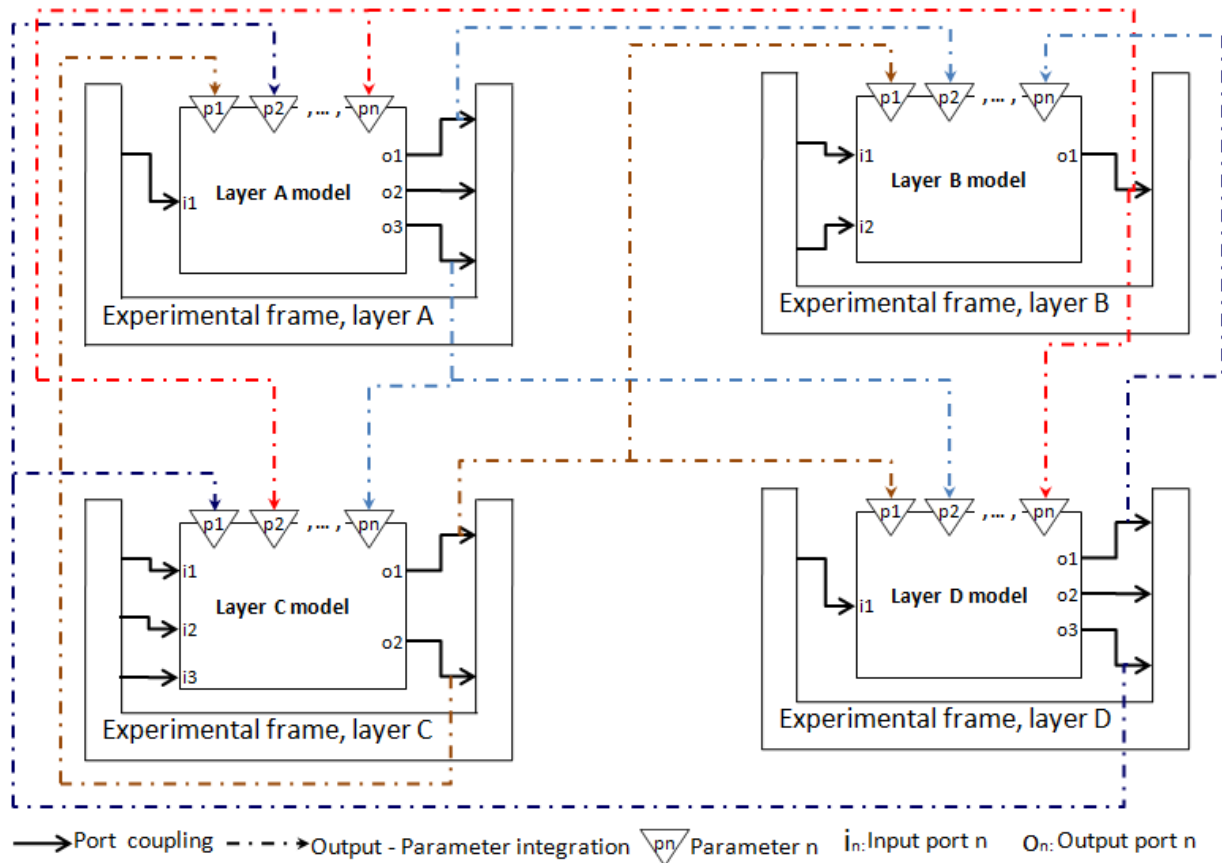


Figure 2. Output-to-parameter integration of HS perspectives.

A DEVS-BASED FORMALISM FOR INTEGRATING HS PERSPECTIVES

This section presents a formalism based on DEVS to describe the output-to-parameter integration of different perspectives of HS as proposed in Figure 2.

An Overview of DEVS

DEVS (Zeigler et al. 2000) is a system-theoretic simulation formalism for DESs. An atomic DEVS model, AM , has a time base and abstract sets of states, transitions, inputs and outputs to describe system's structure and behaviour.

$$AM = \langle X, Y, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle \quad (1)$$

$X = \{(p, v), p \in IPort \wedge v \in dom(p)\}$: set of input events where $IPort$ is the set of input ports.

$Y = \{(q, v), q \in OPort \wedge v \in dom(q)\}$: set of output events where $OPort$ is the set of output ports.

S : set of discrete states.

$ta: S \rightarrow \mathbb{R}_{0,\infty}^+$: time advance function.

$\delta_{int}: S \rightarrow S$: internal state transition function

$\delta_{ext}: \{(s, e) | s \in S, e \in [0, ta(s)]\} \times X^b \rightarrow S$: external state transition; e is the elapsed time since last transition.

$\delta_{conf}: S \times X^b \rightarrow S$: confluent state transition.

$\lambda: S \rightarrow Y^b$: output function.

AM is, at any time, in a state $s \in S$. The internal, external and confluent transition functions dictate the model's new state as follows:

When the elapsed time $e = ta(s)$ expires before any external event occurs, the system outputs the value $y = \lambda(s)$ and changes to state $\delta_{int}(s)$. If an external event $x \in X$ occurs before the expiration of $ta(s)$, the system changes to a state $\delta_{ext}(s, e, x)$; if the event $x \in X$ coincides with the expiration of $ta(s)$, the system outputs the value $y = \lambda(s)$ and changes to a state $\delta_{conf}(s, x)$. Note that outputs are only possible just before internal or confluent transitions. In any case, the system assumes a new state s' with some new elapsed time defined by $ta(s')$.

A coupled DEVS model, CM , is an hierarchical composition of atomic and/or coupled DEVS models to build more complex systems in which the components interact via the exchange of messages between designated input/output ports.

$$CM = \langle X, Y, D, \{M_d\}_{d \in D}, EIC, EOC, IC \rangle \quad (2)$$

X and Y are as defined for AM .

D : set of component names; $\forall d \in D, M_d$ is the full specification represented by d .

$EIC = \{((M, ip_M), (d, ip_d)) | ip_M \in IPorts_M, ip_d \in IPorts_d\}$: set of couplings between the input ports of CM and those of some of its components.

$EOC = \{(d, op_d), (M, op_M) \mid op_M \in OPorts_M, ip_d \in OPorts_d\}$: set of couplings between the output ports of CM and those of some of its components.

$IC = \{(a, op_a), (b, ip_b) \mid op_a \in OPorts_a, ip_b \in IPorts_b\}$: set of couplings between the output and input ports of different components of CM . DEVS and its operational semantics are described in detail in (Zeigler et al. 2000).

A Parameterized DEVS Formalism for Loose Integration of Multi-perspective HS Simulation Processes

This paper proposes a DEVS-based formalism called *Parameterized Atomic DEVS (PAD)* to model different layers of HS with dynamic parameters that represent the information required from concurrent simulations of other layers such that the parameters get live updates from appropriate sources at runtime. The proposed PAD is not another simulation formalism in itself; rather, it maintains DEVS and its simulation protocols but builds on it, a mechanism for realizing live update of parameters in concurrent simulation processes as described in Figure 2. Like DEVS, it is also defined at atomic and network levels.

$$PAD = \langle X, Y, \mathbf{P}, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle \quad (3)$$

PAD introduces an element, \mathbf{P} , to the original atomic DEVS definition in (1) where $\mathbf{P} = \{(p_i, dom(p_i)) \mid i \in \mathbb{N}\}$ is a set of pairs of dynamic parameters, p , and their domains $dom(p)$. Conventionally, the set S of states in DEVS can contain state variables whose instantaneous values determine the states of the system. Hence, changes in the values of state variables are essentially due to reconfigurations during state transitions. In contrast, values of the elements of \mathbf{P} are not affected by the system's internal processes though they may be used for computing new values of state variables, output events and time advances. Rather, elements of \mathbf{P} update their values upon receiving feedbacks from external sources which, in the context of this paper, are models of other perspectives of the HS. The introduction of \mathbf{P} has become necessary in this case since this concept is not represented in the original DEVS.

An earlier solution proposed by Seck and Honig (2012) is to represent the dynamic parameters as input ports that are coupled to the respective sources of live feedbacks. This approach is not suitable in the context of this paper as the reception of an update at such an input port will trigger an undesired external state transition. Changes in the value of $p \in \mathbf{P}$ do not imply changes of state; they only ensure that p is always up to date. Since the parameters are used in computations, the set \mathbf{P} is introduced into the domains of the behavioral functions of PAD as follows:

$$\delta_{int}: S \times \mathbf{P} \rightarrow S \times \mathbf{P}, \delta_{ext}: Q \times X^b \times \mathbf{P} \rightarrow S \times \mathbf{P}, \delta_{conf}: S \times X^b \times \mathbf{P} \rightarrow S \times \mathbf{P}, \lambda: S \times \mathbf{P} \rightarrow Y^b$$

and $ta: S \times \mathbf{P} \rightarrow \mathbb{R}_{0,\infty}^+$

Hence, the formalism is still consistent with the original DEVS and can be simulated using existing simulation packages with minimal additional efforts.

The mechanism of communication among different perspectives of HS is defined as a network, N , of PAD s sending live updates to one another via automated transmission of outputs from each PAD to its representative parameter(s) in others.

$$N = \langle G, \{M_g\}_{g \in G}, \Sigma \rangle \quad (4)$$

$$\Sigma = \left\{ \left(((i, \alpha), (j, \beta)), \theta_{\alpha\beta} \right) \mid i, j \in G, \alpha \in OPorts_i, \beta \in P_j, \theta_{\alpha\beta}: dom(\alpha) \rightarrow dom(\beta) \right\} \quad (5)$$

In (4), G is the set of names of PAD s in the different layers that communicate with one another and every $g \in G$ refers to a complete PAD specification denoted by M_g . The integration set, Σ , is a relation that maps output ports of a PAD to the dynamic parameters of other PAD (s) in other layers of HS. Mathematically, Σ (5) is a set of 2-tuples where the first (resp. second) entry of each element represents the output port-parameter mapping (resp. output-parameter transformer). The output port-parameter mapping (i.e., $((i, \alpha), (j, \beta))$) is also an ordered pair of pairs; the first pair points to a PAD and one of its output ports while the other pair points to a PAD and one of its parameters. Thus, the output port in the former updates the parameter in the latter. The output-parameter transformer function, $\theta_{\alpha\beta}: dom(\alpha) \rightarrow dom(\beta)$, maps the domain of the output port to that of the parameter for each specific output port-parameter mapping. In essence, the transformer specifies the instantaneous operation that may be required to convert output values from α to legal values in the domain of β .

It is important to note that the integration described in (4) and (5) is different from DEVS' coupling mechanism. While the latter establishes connections between input/output ports of components of the same complex system for the purpose of exchanging events and, as consequence, excitation, the former creates links between output ports and parameters of components of isolated systems for the purpose transmitting live updates to the parameters. Therefore, (4) can be re-written using Coupled DEVS formalism as:

$$N = \langle X, Y, G, \{M_g\}_{g \in G}, EIC, EOC, IC, \Sigma \rangle \quad (6)$$

$X = Y = EIC = EOC = IC = \{\}$ and G, M_g and Σ are as described previously in (4) and (5).

Since this integration mechanism is not part of conventional DEVS formalism and its operational semantics, the next subsection presents a guide to implement the parameter updates mechanism as a patch for use with existing DEVS-based simulation packages.

Semantics of PAD

The implementation of the live update mechanisms of PAD presented in this subsection is based on the object-oriented observer and command patterns (Gamma et al. 1994); it is fair to provide an overview of the patterns to put the reader in perspective. Object-Oriented design patterns are reusable solutions to some general and recurrent software engineering problems that can be adapted to suitable contexts to model solutions to contextual problems.

Observer design pattern

The observer design pattern, a behavioral pattern, is "a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically" Gamma et al. (1994).

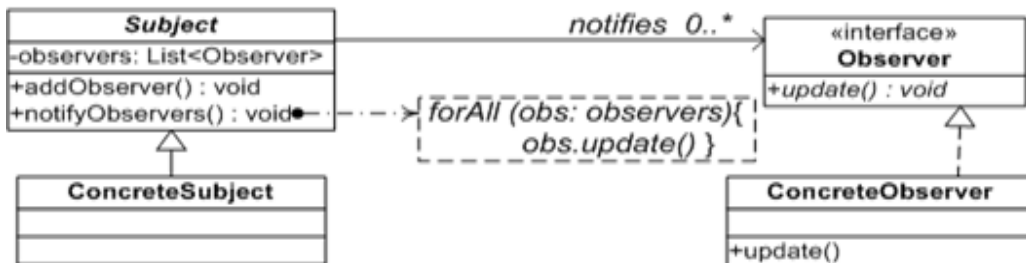


Figure 3. Observer design pattern.

Figure 3 illustrates the pattern. *Subject* maintains a finite list of references to *Observer* objects that are constantly monitoring its (subject's) state. Whenever its state changes, subject *notifies* all its observers in a loop via their *update* methods. Each concrete observer implements the update method according to its own reaction to the notifications. This pattern is widely used in graphical user interface programming and it provides the underlying principle for the Model-View-Controller (MVC) architecture so that all views are automatically updated whenever there is a change of state in the model.

Command design pattern

The *command* design pattern (Gamma et al. 1994) provides a methodology to encapsulate a command (method call) in an object and issue it in such a way that the requested operation and the requesting object do not know each other. In Figure 4, *Client* and the *action()* method of *Receiver* are the requesting object and requesting object and requested operation respectively.

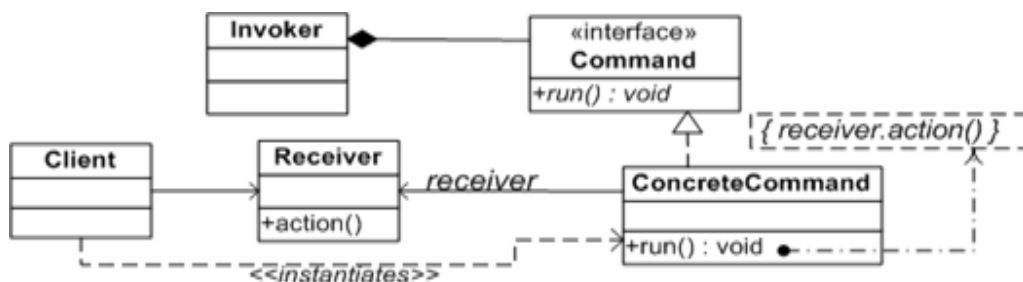


Figure 4. Command design pattern.

Client delegates the invocation to a *command* from a pool of request commands managed by the *Invoker*. The command runs by identifying and invoking the requested operation. Command pattern provides a methodology for asynchronous (non-blocking) method call and has been used to decouple clients from server methods in Asynchronous Remote Method Invocation (Raje et al., 1997). It will be used later in this section to realize asynchronous update of dynamic

parameters in PADs so that simulation processes sending updates are not blocked by the parameter update operations.

Dynamic update/Feedback framework of PAD

Figure 5 presents the framework to implement the output-parameter integration in PAD. The dynamic parameter update mechanism is described based on observer pattern within the dashed box.

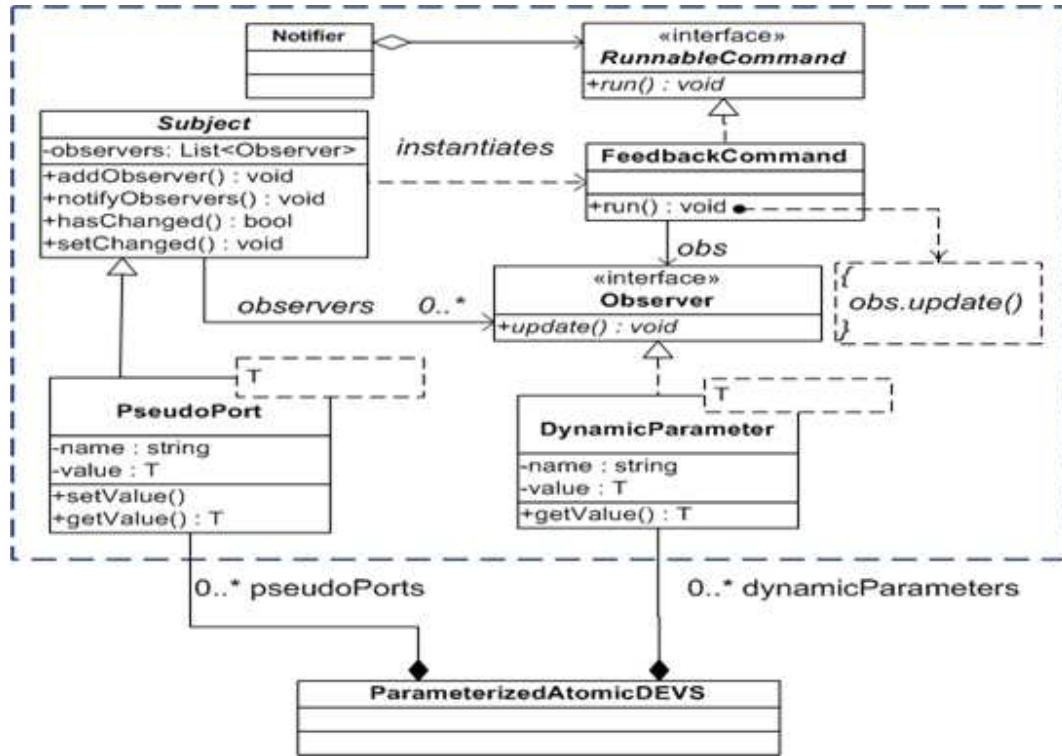


Figure 5. Semantics framework of PAD.

The authors define two generic classes *DynamicParameter* (resp. *PseudoPort*) which extend the *Observer* interface (resp. *Subject* class). Objects of *DynamicParameter* can be created to define parameters while substituting the generic type *T* with the appropriate domain (type) of the declared parameter. Similarly, instances of *PseudoPort* are created to emulate each DEVS output port that needs to send updates to some parameters.

The mapping of outputs to parameters as described in (4) - (6) is actualized at the Coupled DEVS level by adding all dynamic parameters that depend on a pseudo port to its list of observers. The idea here is that whenever there is an output in the simulation of an HS perspective, the same value is automatically transmitted, through the pseudo port attached to the output port concerned, to the appropriate parameters. This is made possible by the semantics of the observer pattern. i.e., the change in the value of the pseudo port (a subject) results in the execution of its *notifyObservers()* method, and consequently, the transmission of the new value to all observing parameters. Unfortunately, *subject* will naturally notify its observers in

sequential blocking method calls. The impending shortcomings of this is that the simulation process sending updates will have to wait till all parameters in target perspectives are updated before it can continue. This problem can be removed by using the command design pattern to achieve asynchronous notification process. This is illustrated in Figure 5 as *PseudoPort* (through *Subject*) delegates the invocation of the *update* method of each *DynamicParameter* to a *FeedbackCommand* as described previously in the *command* pattern. This design can be implemented in general purpose programming languages as an API/plugin to be used with exiting DEVS simulators in the same languages. The authors have done an implementation in Java which is used with a Java-based DEVS simulator package to implement the case study to be presented in the next section.

CASE STUDY

This section presents an example (in Figure 6) to demonstrate the proposed PAD for multi-perspective M&S of HS. Simulation models of the four HS perspective are chosen from the literature and translated into PAD formalism to realize the output-parameter integration. In layer A, the output, o_1 , of the HPD (disease spread) model updates the parameters of the other three layers as shown in the diagram while the parameter of HPD itself receives live updates from the outputs of the AHR model in layer D.

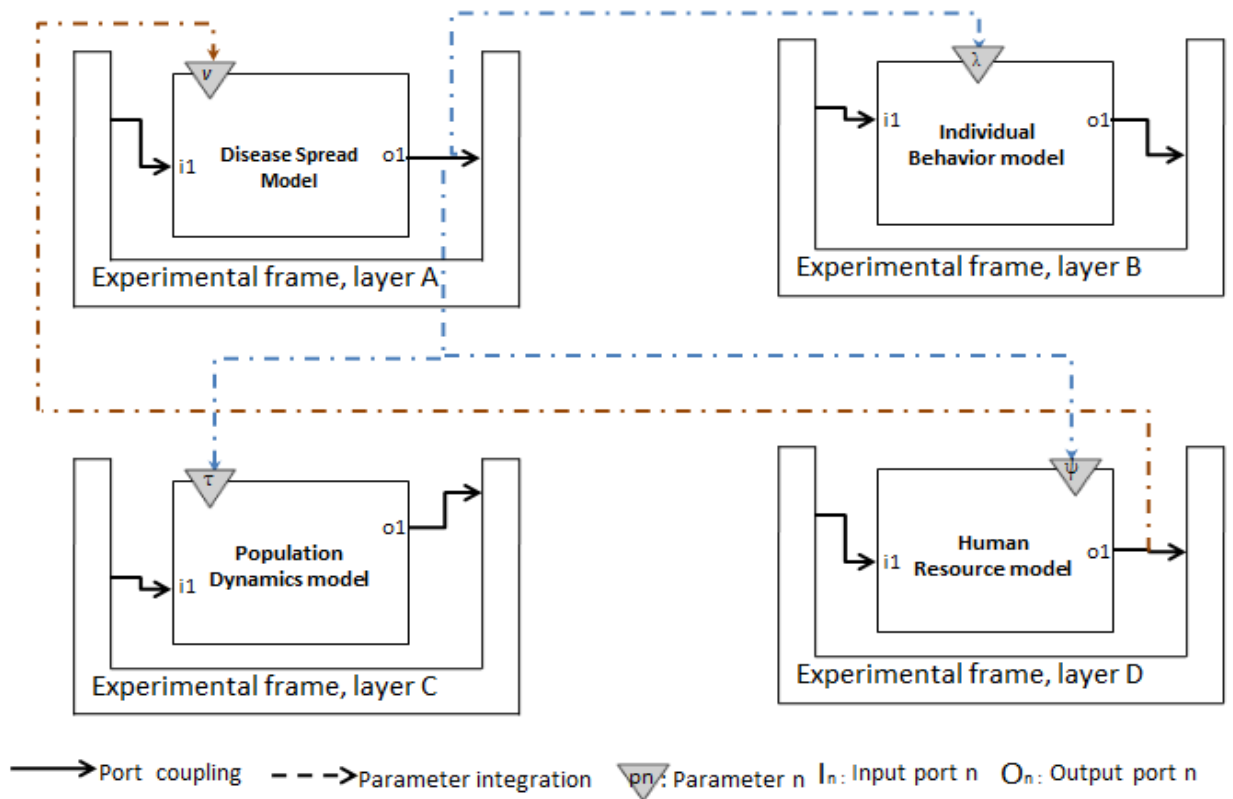


Figure 6. Parameterized DEVS Coupled model

Sequel to equations (4) and (5), the network of PADs in Figure 6 can be described as:

$N = \langle G, \{M_g\}_{g \in G}, \Sigma \rangle$ where $G = \{HPD, IB, PD, AHR\}$ and the integration set Σ is:

$\Sigma = \{(((HPD, O_1), (IB, \mu)), (\mu = 10*b) \mid b \in \text{dom}(O_1)), (((HPD, O_1), (PD, \tau)), (\tau = 3.5*a) \mid a \in \text{dom}(O_1)), (((HPD, O_1), (AHR, \psi)), (\psi = \log_7*c) \mid c \in \text{dom}(O_1)), (((AHR, O_1), (HPD, m)), (m = 1.5*d) \mid d \in \text{dom}(O_1))\}$.

The set has four elements each specifying the update mechanism of one parameter in the diagram. For instance, the first element, $((((HPD, O_1), (IB, \mu)), (\mu = 10 * b) \mid b \in \text{dom}(O_1)))$, specifies the transmission of the outputs b from port O_1 of HPD to update the parameter μ of IB with a transformer operation $\mu = 10 * b$.

The PAD M_g for each $g \in G$ will be described, followed by discussions of the simulation results, under the following sub-headings while their mathematical specifications are presented in the Appendix.

AHR Model

The AHR model in Figure 6 (layer D) reuses the work of Perez and Ntamo (2010), a DEVS-based simulation model to study the management and allocation of human resources to deliver specialized nuclear medicine services. The model particularly describes the transitions of a nuclear medicine technologist, TECH, between different working states - "idle", "get_schedule", "update_schedule", "waiting", "travel_to", "serve_patient", "wait_here", "travel_from" - to render specialized healthcare services. In the original model of Perez and Ntamo, the time advance of "idle" state (i.e., idle time of TECH) is obtained from a constant parameter ψ throughout the simulation; the authors of this paper argue that in reality, it is most likely that the value of ψ varies due to, inter alia, the instantaneous number of patients waiting for TECH's service. This waiting queue can be significantly affected by the HPD in the environment of the healthcare facility. Hence the model is reused in this paper by translating it to a PAD with ψ as dynamic parameter that gets live updates from the output of HPD model in layer A.

HPD Model

The HPD model in Figure 6 (layer A) reuses the work of White et al. (2009), Cellular Automata (CA) model to simulate an epidemic in an environment. The state variable of each cell of the CA takes real values in the range $[0, 1]$ and it is computed based on a local transition function, the current states of its eight neighboring cells and that of the cell itself at each simulation step. The transition function uses three real constants m , v , and c as coefficients of movement of population between cells, connection between the cells and virulence of the disease respectively. The authors of this paper argue that these coefficients will most likely not remain constant in real situation; for instance, the parameter v , virulence of the disease will eventually change due to the impact of health services like vaccination during an outbreak. Using the CA-to-DEVS translation techniques proposed by Wainer (2009), the model is translated into a PAD with a dynamic parameter v , which gets live updates from the AHR model (layer D).

IB Model

This is adapted from the work of Salimifard et al. (2013), a model based on colored Petri Nets (PN) to study patients' flow through different service stages at the emergency unit of a hospital.

They represented the patients' arrival rates by a constant parameter. Again, the authors of this paper opine that the rate may vary dynamically, in real situation, with factors determined by the HPD in that environment. The PN-to-DEVS translation method proposed by Jacques and Wainer (2002) was used to express the model in PAD with a dynamic parameter λ that gets live updates from the HPD model in layer A.

PD Model

The PD model in Figure 6 (layer C) was adapted from (Ng et al., 2011), a system dynamics model disaggregated into three stocks of age groups: 0-14, 15-64, and 65+ for a demographic study of an environment with focus on birth, aging and mortality rates. While the original model maintains a constant mortality rate to simulate the evolution of the different age groups, this paper explores a situation where the mortality may vary dynamically health phenomena like epidemic. Hence the model was translated into a PAD with a dynamic parameter τ representing the mortality rate and which gets live updates from the HPD model in layer A.

Results and Discussions

The simulation of the network of PADs in Figure 6 was carried out using a Java implementation of the DEVS simulation algorithm and a patch implementing the PAD integration framework presented in Figure 5. Due to space constraints, details of the simulator and simulation codes are not provided in this paper; the reader is invited to (Sarjoughian and Zeigler, 1998) for an example of DEVS simulator.

Each PAD was simulated in isolation with constant parameter for 60 simulation time units, then the integrated PADS were simulated concurrently for the same period and allowed to send updates to appropriate parameters. The two results for each of the four HS perspectives will be compared under the following sub-subheadings.

AHR simulation results

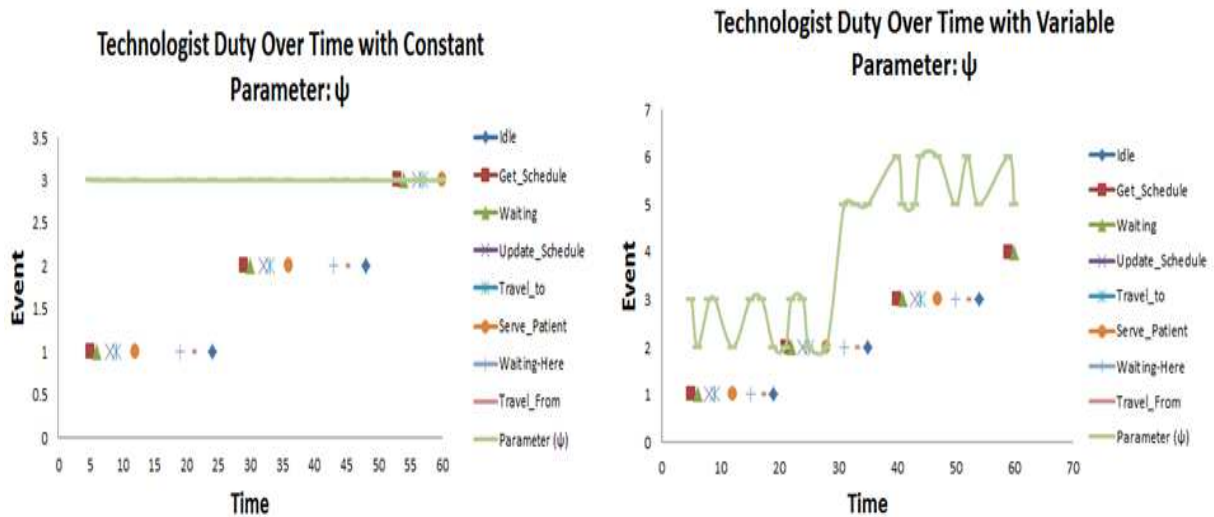


Figure 7. Simulation traces of AHR models

Figure 7 presents the simulated activities of the technologist, TECH, in AHR over time with constant ψ (shown as horizontal line at Event = 3) and dynamic ψ (shown as curve fluctuating between 2 and 6) on the left and right graphs respectively. In either case, TECH visits eight sequential states starting from "Get_schedule" through "serve_patient" to "idle" in the process of treating every service request. We see from the two graphs that, with the constant ψ , TECH takes about 20 time units to service a request and he could service less than three requests within the chosen simulation period. On the contrary, when ψ is left to adjust based on environmental forces, he was able to optimize his throughput, service every request within 15 time units and has started servicing the fourth request at the end of the simulation period. The dynamic ψ graph can give insights into TECH's optimal performance under varying load densities due to environmental influences unlike the constant ψ graph that assumes a uniform load density throughout the simulation.

HPD simulation results

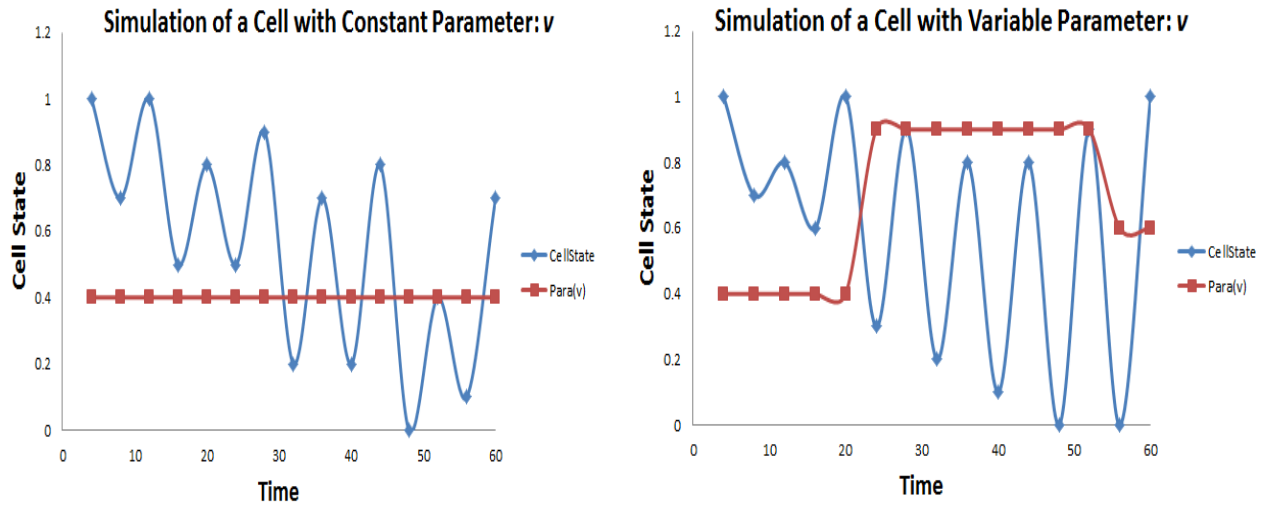


Figure 8. Simulation traces of HPD models

Figure 8 presents the simulation results of the HPD model with constant parameter v (in red) on the left graph and dynamic v on the right graph. The curve (blue color) represents the percentage of the population that is infected during the outbreak. In the graph at the right side the sudden increase of the parameter (v) from the simulation time (20) to simulation time (55) causes the disease curve to maintain its maximum as high as at 0.9 (cell state) while in the graph at left side the maximum can be as low as 0.4 (cell state) within the same range of simulation time. That is, the more the virulence increases the more the people are infected. However, the value of cell state with dynamics parameter that remains under 0.9 is due to some medical interventions in controlling the outbreak. This value is being updated by AHR model.

IB simulation results

Figure 9 presents the simulation results of the IB model with constant parameter λ (patient's arrival rate) on the left graph and dynamic λ on the right graph. The places P1, P2, P3 and P4 in

the figure represent the state where each patient may to be exposed there. E.g. Place P1 and P2 represent state for non-urgent patient and state for urgent patient respectively. In the left side graph when the parameter λ (light blue) remains constant, the highest number of patients arriving in the hospital at a time is up to 28 (blue line). Contrary to that, on the right side graph when the parameter λ is being updated by HPD model and begins to fluctuate the result shows significant increase in the number of patient arriving in the hospital reaching up to 65 at a time. This increase on the arrival is mainly due to the propagation of the disease causing more and more people to seek for healthcare services.

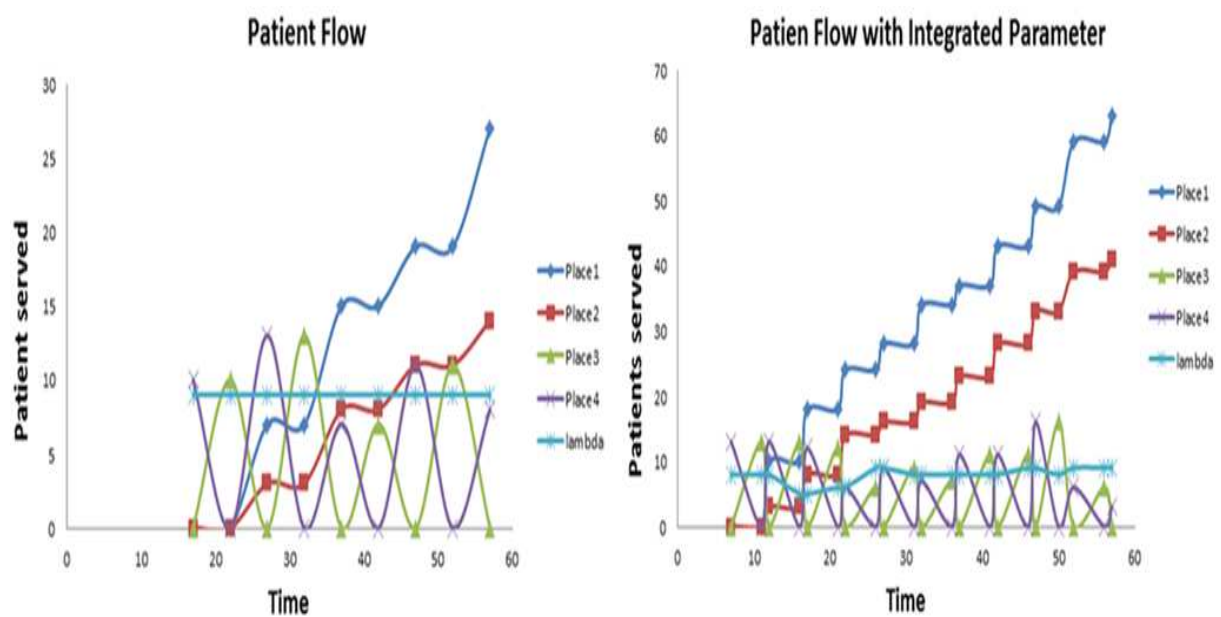


Figure 9. Simulation traces of IB models

PD simulation results

Figure 10 shows the results of simulating the PD model with constant τ (mortality rate) on the left and dynamic τ on the right graph. There is a significant rise in the level of τ in the graph on the right side between simulation time 20 and about 42 due to the accelerated spread of a disease during this period (see our previous discussions on Figure 8); expectedly, this raised the stock of deaths to about 40 during this period and above 50 towards the end of the simulation unlike in the left graph where the stock of deaths did not go beyond 40 throughout the simulation. Though not explicitly stated, the sudden rise in τ appears to affect the active age group 15-64 the most with its stock dropping significantly below 10 during this period and recovering to about 48 towards the end of the simulation; the result in the left graph is slightly different for this age group. The age group 65+ is second most affected as its stock is kept at about 10 throughout the simulation; this value is just about half of what is reported in the left graph.

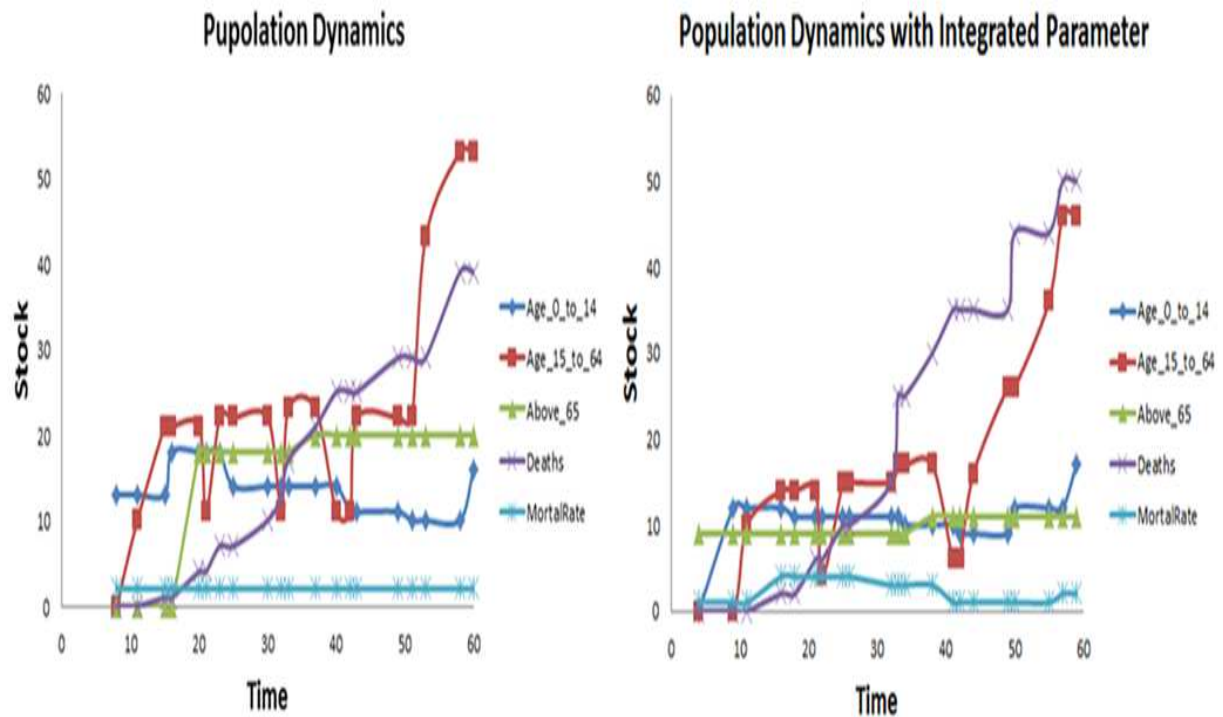


Figure 10. Simulation traces of PD models

Validation

A simulation model is considered to be valid if its accuracy is within the purpose and the question it is designed to answer (Sargent, 2003). The authors considered in this paper, the conceptual model validation which ensures the face validity of the model and its underlying logic. A battery of model tests was conducted with domain experts' opinion on model parameter variations to ensure that the results obtained are closer to the reality and help to document more realistic healthcare policies. The simulation models representing the four perspectives of HSs were run under two scenarios observation. Firstly, the model parameters were maintained constant throughout the simulation. Secondly the model parameters were dynamically updated in a concurrent simulation to reflect the interactions between health phenomena. These parameters include virulence of the epidemic, patient arrival time at a clinic, idleness time of a medical staff, and mortality rate. The tests indicated that the simulation results with live update of models output-to-parameter integration during concurrent simulations were "reasonable" for the intended purpose of the model. The authors are in the process of conducting a more complete observational study and data collection to explore other simulation technics. However, these data were not available at the time of the writing of this paper.

CONCLUSIONS AND FUTURE WORK

This paper proposed a methodology for systematic integration of models of different perspectives of HS to allow for live interactions among them during concurrent simulation processes. The novelty of the work is that it reveals the important and subtle intertwinement of

the processes of different HS perspectives, an aspect that is seldom considered in the usual practice of isolated studies of the different perspectives.

The paper extends a previous work on a disciplined stratification of major HS problems studied by simulation into four perspectives: *allocation of healthcare resources*, *health phenomena dynamics*, *population dynamics*, and *individual behavior*. The usual practice is to study problems in each perspective in isolation with some constant parameters representing the influences of other perspectives; the authors took a slight departure from this approach arguing that, in reality, processes of all perspectives run concurrently and influence one another continuously. They proposed a software engineering mechanism to integrate the simulation processes, based on DEVS formalism, of the four perspectives at runtime through such that each perspective provides live updates of its representative parameter(s) in other perspectives without disturbing their internal processes. They presented a case study to demonstrate the effects of the integrated simulation approach in comparison to the results of isolated simulations of each perspective. A conceptual model validation was presented. Furthermore, the authors intend to proceed on data collection for historical data validation. It would also be interesting to apply this approach to simulation-based studies of non-healthcare domains like public traffic, defense, emergency evacuations, etc.

APPENDIX

The appendix presents the specifications of the PADs of the HS perspectives presented in the case study (Figure 6).

HPD Specification

$$\text{HPD}_{\text{PAD}} = \langle X, Y, P, S, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{con}}, \lambda, \text{ta} \rangle$$

$$X = \{ \text{IN}_0 \in \mathbb{R}, \text{IN}_1 \in \mathbb{R}, \text{IN}_2 \in \mathbb{R}, \text{IN}_3 \in \mathbb{R}, \text{IN}_4 \in \mathbb{R}, \text{IN}_5 \in \mathbb{R}, \text{IN}_6 \in \mathbb{R}, \text{IN}_7 \in \mathbb{R} \}$$

$$Y = \{ \text{OUT} \in \mathbb{R} \}$$

$$P = \{ v \in \mathbb{R} \}$$

$$S = \{ \text{CurrentState}, \text{Computing} \} \times \mathbb{R}$$

$$\delta_{\text{ext}} (((\text{CurrentState}, \Theta), e), x, p) \rightarrow ((\text{Computing}, 0), p)$$

$$\forall \Theta \in \mathbb{R}, \forall x \in X^b, \forall p \in P, e \in [0, \text{ta}(s)]$$

$$\delta_{\text{conf}} ((\text{phase}, \Theta), x, p) \rightarrow \delta_{\text{ext}} (\delta_{\text{int}} (\text{phase}, \Theta), 0, x, p)$$

$$\forall \text{phase} \in \{ \text{CurrentState}, \text{Computing} \}$$

$$\delta_{\text{int}} (((\text{Computing}, \Theta), p) \rightarrow ((\text{CurrentState}, \infty), p)$$

$$\forall \Theta \in \mathbb{R}, \forall p \in P$$

$$\lambda ((\text{Computing}, \Theta), p) \rightarrow v$$

$$\forall v \in \mathbb{R}, \forall \Theta \in \mathbb{R}, \forall p \in P$$

$$\text{ta} ((s, \Theta), p) \rightarrow \Theta$$

$$\forall s \in S, \forall \Theta \in \mathbb{R}, \forall p \in P.$$

IB Specification (PAD for Place)

$IB_{PAD} = \langle X, Y, P, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$

$X = \{IN \in \mathbb{N}\}$

$Y = \{OUT \in \mathbb{N}\}$

$P = \{\lambda \in \mathbb{R}\}$

$S = \{\text{Receiving}, \text{Sending}\} \times \mathbb{N} \times \mathbb{R}$

$\delta_{ext}: (((s, n, \Theta), e), x, \lambda) \rightarrow ((s, n + x, 0), \lambda)$

$\forall s \in S \wedge s \neq \text{sending}, \forall n \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall x \in X^b, \forall \lambda \in P, e \in [0, ta(s)]$

$\delta_{ext}: (((s, n, \Theta), e), x, \lambda) \rightarrow ((s, n, \Theta - e), \lambda)$

$\forall s \in S \wedge s = \text{sending}, \forall x \in X^b, \forall n \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \lambda \in P, e \in [0, ta(s)]$

$\delta_{con} ((\text{phase}, n, \Theta), x, \lambda) \rightarrow \delta_{ext} (\delta_{int} (\text{phase}, n, \Theta), 0, x, \lambda)$

$\forall \text{phase} \in \{\text{Receiving}, \text{Sending}\}$

$\delta_{int} ((\text{Sending}, n, \Theta), \lambda) \rightarrow ((\text{Receiving}, n, \infty), \lambda)$

$\forall n \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall \lambda \in P$

$\lambda ((\text{Sending}, n, \Theta), \lambda) \rightarrow n$

$\forall n \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall \lambda \in P$

$ta ((s, n, \Theta), \lambda) \rightarrow \Theta$

$\forall s \in S, \forall n \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall \lambda \in P$

PD Specification

$PD_{PAD} = \langle X, Y, P, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$

$X = \{IN_0 \in \mathbb{N}\}$

$Y = \{OUT \in \mathbb{N}\}$

$P = \{\tau\}$

$S = \{\text{Idle}, \text{Updating}\} \times \mathbb{N} \times \mathbb{R}$

$\delta_{ext}: (((\text{Idle}, \text{stock}, \Theta), e), x, p) \rightarrow ((\text{Updating}, \text{stock}, 0), p)$

$\forall \text{stock} \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall x \in X^b, \forall p \in P, e \in [0, ta(s)]$

$\delta_{con} ((\text{phase}, \text{stock}, \Theta), x, p) \rightarrow \delta_{ext} (\delta_{int} (\text{phase}, \text{stock}, \Theta), 0, x, p)$

$\forall \text{phase} \in \{\text{Idle}, \text{Updating}\}$

$\delta_{int} ((\text{Updating}, \text{stock}, \Theta), p) \rightarrow ((\text{Idle}, \text{stock}, \infty), p)$

$\forall \text{stock} \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall p \in P$

$\lambda ((\text{Updating}, \text{stock}, \Theta), p) \rightarrow v$

$\forall \text{stock} \in \mathbb{N}, \forall \Theta \in \mathbb{R}, \forall p \in P, \forall v \in \mathbb{R}$

$ta ((s, \text{stock}, \Theta), p) \rightarrow \Theta$

$\forall \text{stock} \in \mathbb{N}, \forall s \in S, \forall \Theta \in \mathbb{R}, \forall p \in P$

REFERENCES

- Ahmed, M. A., Alkhamis, T. M. (2009). Simulation optimization for an emergency department healthcare unit in Kuwait. *European Journal of Operational Research*, 198(3), 936-942.
- Barjis, J. (2011). Healthcare simulation and its potential areas and future trends. *SCS M&S Magazine*, 2(5), 1-6.
- Bonnetain, E., Boucheix, J. M., Hamet, M., Freysz, M. (2010). Benefits of computer screen-based simulation in learning cardiac arrest procedures. *Medical education*, 44(7), 716-722.
- Braun, R., Esswein, W. (2015). Towards multi-perspective modeling with BPMN. In *Advances in Enterprise Engineering IX* (pp. 67-81). Springer.
- Bruzzzone, A. G., Frascio, M., Longo, F., Massei, M., Siri, A., Tremori, A. (2012). MARIA: An Agent Driven Simulation for a Web Based Serious Game devoted to Renew Education Processes in Health Care. In *WISHC 2012* (pp. 188-194), 978-88-97999-13-3.
- Carr, S., Roberts, S. (2010). Planning for infectious disease outbreaks: A geographic disease spread, clinic location, and resource allocation simulation. In *WSC 2010* (pp. 2171-2184). IEEE.
- Charfeddine, M., Montreuil, B. (2010). Integrated agent-oriented modeling and simulation of population and healthcare delivery network: application to COPD chronic disease in a Canadian region. In *WSC 2010* (pp. 2327-2339). IEEE.
- Choi, B. K., Kang, D., Kong, J., Kim, H., Jamjoom, A. A., Mogbil, A. M., Alghamdi, T. A. (2013). Simulation-based operation management of outpatient departments in university hospitals. In *WSC 2013* (pp. 2287-2298). IEEE.
- Cote, M. J. (1999). Patient flow and resource utilization in an outpatient clinic. *Socio-Economic Planning Sciences*, 33(3), 231-245.
- Dibble, C. (2010). Effective real-time allocation of pandemic interventions. In *WSC 2010* (pp. 2211-2220). IEEE.
- Djitog, I., Aliyu, H. O., Traoré, M. K. (2015). Towards a Framework for Holistic Analysis of Healthcare Systems. In *ESM 2015* (pp. 193-196), *EUROSIS-TI, ISBN: 978-90-77381-90-8*.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). *Design Patterns: Elements of Object-Oriented Software Architecture*. Addison-Wesley, 9, 12.
- Günel, M. M., Pidd, M. (2010). Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, 4(1), 42-51.
- Harper, P. R. (2002). A framework for operational modelling of hospital resources. *Health care management science*, 5(3), 165-173.
- Jacques, C. J., Wainer, G. A. (2002). Using the CD++ DEVS toolkit to develop Petri Nets. In *SummerSim 2002* (pp. 51-56). SCS International.
- Kasaie, P., Dowdy, D. W., Kelton, W. D. (2013). An agent-based simulation of a tuberculosis epidemic: understanding the timing of transmission. In *WSC 2013* (pp. 2227-2238). IEEE.
- Khurma, N., Salamati, F., Pasek, Z. J. (2013). Simulation of patient discharge process and its improvement. In *WSC 2013* (pp. 2452-2462). IEEE.
- Kingston, J. (2001). Ontologies, multi-perspective modelling and knowledge auditing. In *KI 2001*.

- Kuhl, M. E. (2012). A simulation study of patient flow for day of surgery admission. In WSC 2012 (pp. 1-7). IEEE.
- Mikulecky DC (2001) The emergence of complexity: science coming of age or science growing old? *ComputChem* 25(4):341–348
- Ng, A. T. S., Sy, C., Li, J. (2011). A system dynamics model of Singapore healthcare affordability. In WSC 2011(pp. 1-13). IEEE.
- Okhmatovskaia, A., Buckeridge, D. L., Shaban-Nejad, A., Sutcliffe, A., Finès, P., Kopec, J. A., Wolfson, M. C. (2012). Simpho: an ontology for simulation modeling of population health. In WSC 2012 (pp. 1-12). IEEE.
- Ozcan, Y. A., Tànfani, E., &Testi, A. (2011). A simulation-based modeling framework to deal with clinical pathways. In WSC 2011 (pp. 1190-1201), IEEE.
- Pérez, E., Ntamo, L., Bailey, C., McCormack, P. (2010). Modeling and simulation of nuclear medicine patient service management in DEVS. *Simulation* 86(8-9), 481-501.
- Raje, R. R., Williams, J. I., Boyles, M. (1997). Asynchronous remote method invocation (ARMI) mechanism for Java. *Concurrency - Practice and Experience*, 9(11), 1207-1211.
- Reineke, J., Tripakis, S. (2014). Basic problems in multi-view modeling. In TACAS 2014 (pp. 217-232). Springer.
- Roberts, S. D. (2011). Tutorial on the simulation of healthcare systems. In WSC 2011 (pp. 1408-1419), IEEE.
- Rosen, R. (2000) *Essays on life itself*. Columbia University Press, New York
- Salimifard, K., Hosseini, S. Y., Moradi, M. S. (2013). Improving Emergency Department Processes Using Coloured Petri Nets. In PNSE+ ModPE 2013 (pp. 335-349).
- Sarjoughian, H. S., Zeigler, B. P. (1998). DEVSJAVA: Basis for a DEVS-based collaborative M&S environment. *Simulation Series*, 30, 29-36.
- Seck, M. D., &Honig, H. J. (2012). Multi-perspective modelling of complex phenomena. *Computational and Mathematical Organization Theory*, 18(1), 128-144.
- Shin, S. Y., Balasubramanian, H., Brun, Y., Henneman, P. L., Osterweil, L. J. (2013). Resource scheduling through resource-aware simulation of emergency departments. In WSEHC 2013 (pp. 64-70). IEEE.
- Tekinay, Ç., Seck, M., Fumarola, M., Verbraeck, A. (2010). A context-based multi-perspective modeling and simulation framework. In WSC 2010 (pp. 479-489), IEEE.
- Topaloglu, S. (2006). A multi-objective programming model for scheduling emergency medicine residents. *Computers & Industrial Engineering*, 51(3), 375-388.
- Vangheluwe, H. L. (2000). DEVS as a common denominator for multi-formalism hybrid systems modelling. In CACSD 2000 (pp. 129-134), IEEE.
- Wainer, G. A. (2009). *Discrete-event modeling and simulation: a practitioner's approach*. CRC Press.
- Weng, S. J., Tsai, B. S., Wang, L. M., Chang, C. Y., Gotcher, D. (2011). Using simulation and data envelopment analysis in optimal healthcare efficiency allocations. In WSC 2011 (pp. 1295-1305), IEEE.

White, S. H., del Rey, A. M., Sanchez, G. R. (2009). Using cellular automata to simulate epidemic diseases. *Applied Mathematical Sciences*,3(20), 959-968.

Zeigler, B. P., Carter, E., Seo, C., Russell, C. K., Leath, B. A. (2012). Methodology and modeling environment for simulating national health care. In *AutumnSim 2012*.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press.

Sargent, R. G. (2003). Verification and validation of simulation models. In *WSC 2005* (pp. 130-143), IEEE.