Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

AN ACTOR BASED SIMULATION DRIVEN DIGITAL TWIN FOR ANALYZING COMPLEX BUSINESS SYSTEMS

Souvik Barat

Tata Consultancy Services Research
Hadapsar Industrial Estate
Pune 411013, INDIA

Vinay Kulkarni

Tata Consultancy Services Research
Hadapsar Industrial Estate
Pune 411013, INDIA

Tony Clark

Aston University
Aston St
Birmingham B4 7ET, UK

Balbir Barn

Middlesex University London
The Burroughs, Hendon
London NW4 4BT, UK

ABSTRACT

Modern enterprises aim to achieve their business goals while operating in a competitive and dynamic environment. This requires that these enterprises need be efficient, adaptive and amenable for continuous transformation. However, identifying effective control measures, adaptation choices and transformation options for a specific enterprise goal is often both a challenging and expensive task for most of the complex enterprises. The construction of a high-fidelity digital-twin to evaluate the efficacy of a range of control measures, adaptation choices and transformation options is considered to be a cost effective approach for engineering disciplines. This paper presents a novel approach to analogously utilise the concept of digital twin in controlling and adapting large complex business enterprises, and demonstrates its efficacy using a set of adaptation scenarios of a large university.

1 INTRODUCTION

The economic significance of a digital twin (Boschert and Rosen 2016) to design, develop and control a system, subsystem and business process is a well established fact in engineering disciplines (Schleich et al. 2017) and mission critical systems (Glaessgen and Stargel 2012). In such cases, they develop high-fidelity physics and/or mathematical models to represent system behaviour, and use simulation to understand system behaviour and the consequences of various system changes over time. A digital twin, simply put, is a digital representation of a real-world entity or system. Critically, it is a synthetic representation of the structure, behaviour, current state and historical states of an entity for query-based interaction and *what-if* analysis. The key objective of such initiatives is to reduce or eliminate real-life experimentations, which are often expensive or infeasible. It is expected that an effective digital twin expedites the time-to-market of a product (as ineffective design choices can be identified in advance), reduces the cost of system development (due to a reduction of real life experimentations), and helps to arrive at informed control measures (as the consequences of the prospective control measures can be evaluated prior to their implementations). While the benefits and utilisation of a digital twin are well established in some disciplines, its utilisation is not yet in mainstream practice for business enterprises, such as supply chain, business process outsourcing organisations, software service provisioning organisations and operation of large universities. The state of the practice to understand the consequences of prospective changes for controlling and adapting such enterprises are still chiefly driven by human intuition as opposed to any quantitative evidences.

This paper presents an advancement over state-of-the-practice of business enterprise modelling and analysis using – (a) a digital twin based decision evaluation framework for business facing enterprises, and (b) an actor-based modelling abstraction to intuitively represent large and complex enterprises. The proposed approach adopts the following considerations – (i) principle of control system as the basis of the proposed approach (ii) actor-based modelling abstraction (Agha 1986; Hewitt 2010) for constructing digital twin of complex enterprise, (iii) bottom-up simulation technique as an analysis means or an epistemic engine (Tolk et al. 2013), (iv) method proposed by Robert Sergent (Sargent 2005) as methodological rigour for establishing the faithfulness of the constructed digital twin.

We adopt actor-based modelling abstraction as the enterprises are typically known as complex entities with a large number of interdependent subsystems or elements (*i.e.*, system of systems) where elements are *autonomous* and they exhibit significant probabilistic and emerges behaviours (Holland 2006). Therefore, an *actor* abstraction (Agha 1986) that supports modularity, autonomy, reactiveness and emergentism is suitable to closely imitate the real enterprise and its constituent (micro) elements as they exist. The bottom-up simulation of constructed digital twin using interacting actors helps to understand the overall dynamism and emergentism. The introduction of the canonical simulation method and validation techniques proposed by Robert Sargent in actor-based systems, which is a novel contribution, helps to validate the bottom-up formation of complex enterprises and their emergent behaviours. The control theoretic feedback loop within an extended form of the canonical simulation method supports a pragmatic human-in-the-loop observation-based iterative *what-if* analyses for informed decision-making for control and adaptation of complex enterprises. We illustrate our approach using a set of control scenarios of a university that aim to improve the ranking through better research and teaching outcome.

2 ILLUSTRATIVE CASE STUDY

A hypothetical university, we refer to it as ABC University, aims to improve its ranking with respect to its overall academic position and research ranking. The academic ranking is typically measured using the results of the final year students and the student satisfaction score (*i.e.*, an outcome of the students survey on various aspects that includes the quality of feedback to the assessments, the employability options and quality of teaching and facilities). The research ranking is primarily measured based on research outcomes, such as the publication records at top-tier conferences and journals (Altbach 2007).

Influential world ranking tables publish a range of improvement factors or courses of action (Lukman et al. 2010) as potential control or adaptation measures to achieve various university goals. The courses of action which are commonly considered are: (i) academic and student ratio, (ii) balance between research and teaching academics, (iii) work priorities of the academics, (iv) appropriate timetabling, and (v) experience and academic records of the faculty staff. While the potential courses of action are known to the decision makers of a university, such as a Dean and Head of Department, which course of action is the most effective for a university is not known as the effectiveness of these courses of action depends on factors, such as current state of the university, its strengths, quality of potential students. Therefore, those decision makers keep exploring: What alternative or set of alternatives are effective to improve the teaching and research indicators? How do the alternatives compare? Are there any negative consequences of a chosen alternative?

This paper considers a department, let's say Computer Science (CS) department, of ABC University as a running example to illustrate proposed approach and demonstrate its efficacy.

3 TENETS OF COMPLEX ENTERPRISE

An enterprise can be analysed by evaluating various aspects of interest and their characteristics. The Zachman Framework (Zachman 1987) (a foundation for most of the Enterprise Modellings) considers six interrogative aspects: *why*, *what*, *how*, *when*, *where*, and *who* to describe and understand an enterprise.

The General System Theory (GST) (Von Bertalanffy 1968), theory on Complex Adaptive System (CAS) (Holland 2006), complexity theory and organisational theory (Simon 1991) collectively describe the characteristics of a complex enterprise. Organisational theory (Simon 1991) visualises modern enterprise as *reactive* (as it exchanges messages and resources with its environment) and *complex* (consists of interconnected components that work together) system. The foundation of *complexity theory* considers an enterprise as *complex* entity because it often composes a large number of interdependent subsystems or elements in a *nonlinear* way (Casti 1994). Daft *et al.* characterise an enterprise as a composition of multiple *loosely coupled* and *autonomous* elements (Daft and Lewin 1990). The *complex adaptive system* theory considers the behaviour of a complex enterprise is largely probabilistic and emerges from the interactions of the connected sub-systems, individuals or *agents* as opposed to a monolithic deterministic automaton (Holland 2006). Collectively, they emphasise multiple socio-technical characteristics such as: *modularity*, *composability*, *autonomy*, *reactiveness*, *adaptability*, *uncertainty* and *emergentism*.

For example, the CS Department of ABC University comprises a set of academics and students. This structure describes the *what* aspect of an enterprise. It exhibits a specific behaviour that largely relies on the behaviours of the academics and students. The behaviours of these active elements define the *how*, *who*, *when* and *where* aspects. The goals of CS department defines the *why* aspect of an enterprise.

In this example, the CS department, academics and students are *modular* elements. A department can be dynamically formed using varying number of academics and students. Therefore, a department is a *composable* element. The academics and students are *reactive* elements as they can react to the environmental events such as *Call for Paper*. They can continue their activities or interrupt current activity without any external stimuli, therefore they are *autonomous* elements. The department may change the topology over time by adding and/or eliminating academics and students; similarly academics and students may change their behaviour over time - therefore they are *adaptable* elements.

The academics and students exhibit different levels of *uncertainties*. For example, an academic shows probabilistic behaviour to choose research over writing paper at a specific time slot. Similarly, there is an inherent randomness to know an individual student will study in the free hours or not. This case also highlights the emergent behaviour. For example, the behaviour of CS Department typically emerges from the behaviour of constituent academics and students as opposed to a well defined static behaviour.

4 DIGITAL TWIN AND RELEVANT TECHNOLOGY SPECTRUM

A digital twin is a comprehensive and machine-interpretable description of components, behaviours and operation of a real system, process, product or an enterprise for a range of interrogative and predictive analyses and decision making (Grieves and Vickers 2017). Conceptually, a digital twin approach is formed using three core elements: (a) real environment, (b) virtual environment and (c) a two-way connection between the two environments. The information link from real environment to virtual environment serves to: (i) provides information about real environment that helps to construct a virtual environment and ensure its faithfulness, and (ii) supplies system data to set the state of a virtual environment same as a real environment. The reverse information link, *i.e.*, information link from virtual environment to real environment, communicates the effective control instructions and change recommendations that include the change in structure, behaviours and/or goal of a real environment. The real environment is an actual enterprise, system, product or a process. The virtual environment is a faithful representation of a real environment. A virtual environment is formed for a range of interrogative and/or predictive analysis, where the key objectives are: (a) understand real environment in a precise form, (b) analyse the efficacy of various changes or adaptation strategies to realise specific goals of a real environment, and (c) explore design alternatives of a new real environment.

The efficacy of a digital twin chiefly relies on four key factors: (i) construction of a virtual environment (ii) validity of constructed virtual environment, (iii) effective analysis techniques to quantitatively comprehend a virtual environment, and (iv) data sensing and transformation to make virtual environment up-to-date. The existing wide spectrum of modelling, analysis and simulation capabilities help to represent enterprises and support a range of analysis needs. At one extreme, there are mathematical models, such as linear programming (Candes and Tao 2005), that represent systems using mathematical equations and use optimisation techniques for precise analyses and problem solving. However, they are best suited for deterministic and bounded systems. The other class of models are various Enterprise Models (EMs) that range from architectural descriptions, *e.g.* ArchiMate (Iacob et al. 2012), to business process simulations, *e.g.*, BPMN (White 2008), multi-modelling and co-simulations approaches, *e.g.*, DEVS (Camus et al. 2015), or system dynamic models (Forrester 1994) to simulate dynamic behaviour of a system. These modelling and analysis techniques adopt a top-down approach to represent enterprises and use reductionist view for precise analysis. The key limitation of these techniques is that they are not cognisant of inherent emergentism of complex enterprise.

The languages and specifications advocating the *actor model of computation* (Agha 1986) and the agent-based modelling approaches (Macal and North 2010), such as Erlang (Armstrong 1996) and Akka (Allen 2013), support emergentism through bottom-up simulation. They fare better in analysing the systems with socio-technical characteristics such as modularity, autonomy, reactiveness and adaptability. However, they do not support the specification of complex enterprise hierarchies and behavioural uncertainty. Therefore, we argue that an extended form of *actor* based modelling technique that supports complex hierarchical structure and capable of specifying behavioural uncertainty is better suited for representing virtual environment of complex enterprise and their constituent elements. Moreover, we consider actor/agent based simulation an effective analysis aid as it can observe the emergent behaviour through bottom-up simulation.

5 PROPOSED APPROACH

We propose a model-based simulation-driven digital twin framework to evaluate the efficacy of potential control measures, adaptation strategies or transformation choices of complex business enterprises. The proposed framework, as depicted in Figure 1, contains a decision-making unit and two connected control loops – (a) virtual evaluation loop and (b) real evaluation loop. The decision making unit is principally a controller or evaluator that decides a course of action and observes its efficacy using virtual environment and/or real environment. The virtual evaluation loop supports model-based simulation to explore various courses of action (*i.e.*, potential control measures, adaptation strategy and transformation choices) in an environment, which is synthetic but closely represents the reality. The real evaluation loop is an environment for actual experimentation or evaluation of a course of action in reality. As shown in the figure, the connection between virtual evaluation loop and real evaluation loop are threefold – (i) knowledge flow to construct virtual enterprise, which is a faithful purposive representation of real enterprise, (ii) information flow to establish the linkage between virtual goals and real enterprise goals, and (iii) data flow to make virtual enterprise up-to-date with real enterprise. We assume the decision making unit explores all possible courses of action to achieve an enterprise goal using virtual evaluation loop before experimenting them in reality. It is also expected that decision-making unit decides the next possible course of action based on the gathered knowledge about the historical evaluations conducted using virtual evaluation loop and real evaluation loop. Currently we consider human experts who learn from historical evaluations and decide next course of action but the proposed decision making unit can also be realised using sophisticated learning agent, such Reinforcement Learning (RL) agent (Sutton, Barto, et al. 1998).

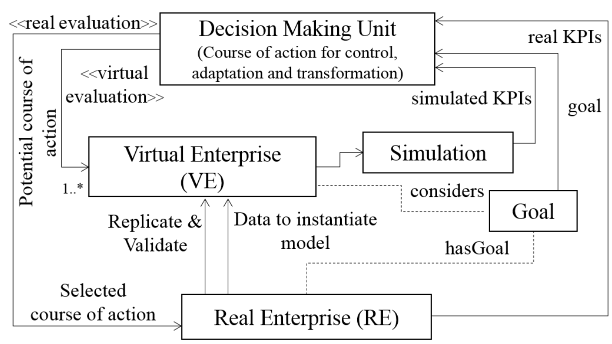


Figure 1: Model Based Digital Twin

In this framework, we consider an actor-based modelling abstraction (Agha 1986; Hewitt 2010) to construct virtual enterprise that closely represents a real enterprise, and an extended form of simulation model construction and validation method proposed by Robert Sargent in (Sargent 2005). In this section,

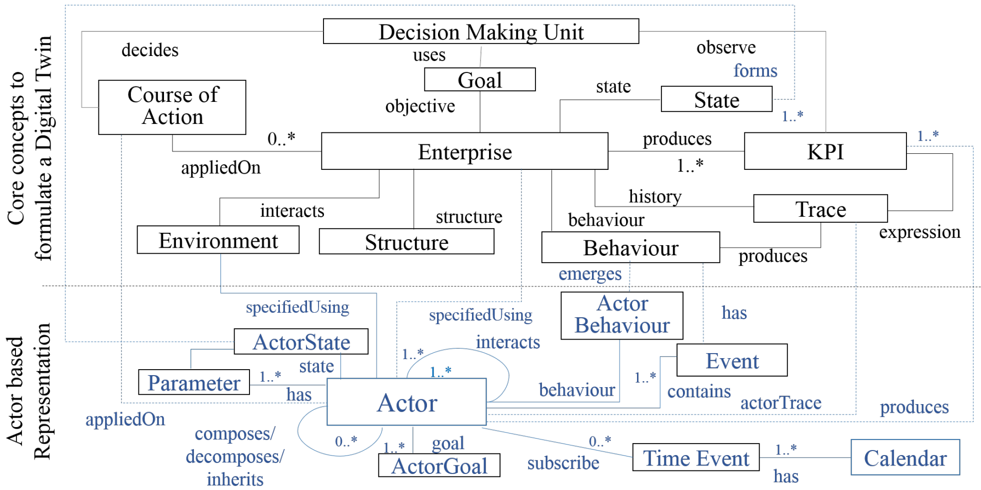


Figure 2: A Meta Model to Represent an Enterprise

we present a meta-model to represent an enterprise using *actor* abstraction and our extended form of model construction and validation method.

5.1 Actor Based Representation

The core concepts to represent an enterprise are depicted using a meta-model as shown in the upper part of the Figure 2. A typical enterprise operates in an *Environment* to achieve its *Goals*. The achievability of the stated *Goals* is determined by a set of key performance indicators or *KPIs*. These *KPIs* are typically computed from the current *State* of an enterprise and/or historical states, *i.e. Trace*. The *State* and *Trace*are chiefly governed by its *Structure* and *Behaviour*. As shown in the figure, the decision makers (*i.e.*, *Decision Making Unit*) assess KPIs with respect to the stated *Goals* to analyse the efficacy of a *Course of Action*.

We use the notion of an *Actor* to represent an enterprise as shown in the lower part of the meta-model shown in Figure 2. Conceptually, an `Enterprise` and and its `Environment` can be represented using a set of `Actors`, where these `Actors` are modular and reactive units. They are characterised by a set of `Parameters` and have their own state, goals, behaviour and trace. They interact with each other using `Events`. The collection of `Actor` specific `ActorStates` form the enterprise `State`. The `Behaviours` of `Actors` collectively define the enterprise `Behaviour` and the aggregation of `Actor` specific fragmented `Traces` form the `Trace` of an `Enterprise`. An `Actor` may own enterprise `KPIs` and enterprise specific `Courses of Action` are chiefly applied to an `Actor` or a set of `Actors`. An `Actor` can be composed or decomposed to any level to imitate an enterprise and their units (*i.e.*, enterprise structure). In addition to the *Events*, which are raised by the constituent *Actors*, there is a concept of *Time Events*, which indicate various time units for simulation.

For an illustration, the ABC University can be visualised as a *University* `Actor`, where the *University* `Actor` is a composition of a set of *Department* `Actors`. Each of these *Department* `Actor` is formed using a set of *Academic* and *Student* `Actors`. Each of these `Actor`, *i.e.*, *University*, *Department*, *Academic* and *Student*, has its own goals and states. The `Behaviour` of a *University* `Actor` is chiefly derived from the `Behaviours` of *Departments*, and `Behaviour` of a *Department* `Actor` is formed using the `Behaviours` of its constituent *Academics* and *Students*. The *Academics* are responsible for the enterprise level *KPIs*, such as number of publications, queries raised by the students and complains raised. The changes of a *University* or a *Department* can be realised by changing the formation and/or behaviour of the *Academics* and/or *Students*, *i.e.* using appropriate `Course of Action` definitions. Constituent *Actors* subscribe to various *Time Events*, such as hour, day, week, month and year.

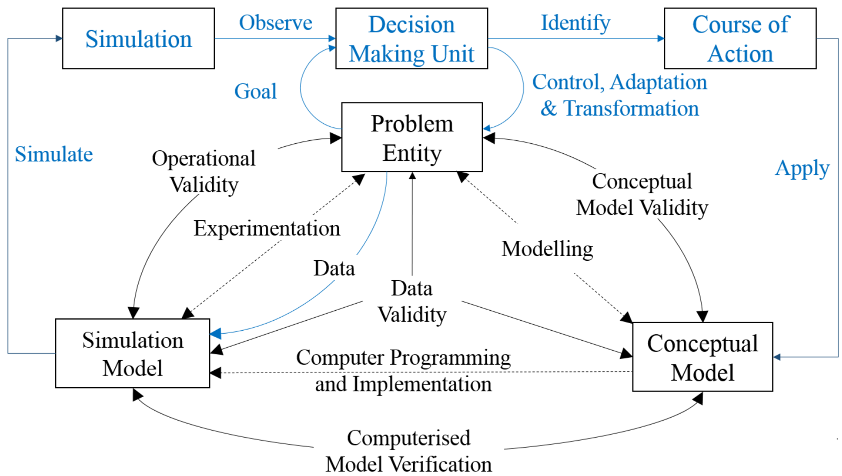


Figure 3: Method to Construct, Validate and Analyse Digital Twins – An Extended Canonical Method

5.2 Model Construction, Validation and Analysis Method

Much of simulation research uses the method proposed by Robert Sargent (Sargent 2005) as highlighted in Figure 3. The essence of the method outlines three representations: *problem entity*, *conceptual model* and *computerised model* or *simulation model* that are derived from each other through an ordered process that can be validated at multiple stages. For example, a conceptual model is validated with respect to the problem entity (*i.e.*, *conceptual model validity*) and *operational validity* ascertains that the simulation results are sufficiently accurate. We adopt this canonical method as a basis to create a digital twin of an enterprise. We consider the problem entity is the real enterprise, and constructed conceptual model and simulation model are two forms of virtual environment. The conceptual model is an intermediate form and simulation model is a virtual and machine-interpretable (simulatable) form of an enterprise. We also consider canonical validation techniques, such as operational validity, as the means to evaluate the faithfulness of the constructed virtual environment.

We extend this canonical method along two dimensions as shown using blue colored lines in the Figure 3 – (i) added a data flow link from problem entity to simulation model to make simulation model up-to-date with respect to the real environment, and (ii) added an overarching loop to observe simulation result or KPIs, evaluate them with respect to enterprise goals, decide the changes or suitable courses of action and apply them on *conceptual model*. This extension helps to explore various *what-if* scenarios using a faithful virtual environment where the faithfulness is ensured through validation techniques suggested in the canonical method. In this extended method, we use class diagram and state machine notation as the specification aids for describing the intermediate conceptual model, and an actor-based simulation language that realises the proposed enterprise meta-model depicted in Figure 2 as the simulation specification. A actor language termed as Enterprise Simulation Language (ESL) (Clark et al. 2017) that realises the proposed actor meta-model described in Figure 2 is considered for simulation specification. However, we argue other existing actor/agent language, such as Akka, can be considered as simulation specification provided the concepts introduced in the proposed actor meta-model are explicitly programmed.

6 DEVELOPING DIGITAL TWIN FOR UNIVERSITY CASE STUDY

In this section, we construct an actor-based digital twin of CS department using the proposed method, and demonstrate how the constructed model helps to analyse potential courses of action prior to their implementation in the reality.

Table 1: Activities of Academics and Students

Key Actor	Activities
Research Academic	Research, Paper Writing, Managerial Work, Unplanned Work (Query Resolution, Complain Resolution)
Teaching Academic	Prepare for Lecture, Deliver Lecture, Prepare for Student Assessment, Assess Student, Unplanned Work
Research and Teaching	The combination of the activities of Research Academic and Teaching Academic
Student	Attend Lecture, Self Study, Appear for Assessment, Raise Query, Raise Complaint

6.1 Problem Entity

CS department of ABC university offers a set of courses to the undergraduate students and focuses on research activities. In particular, teaching academics focus on teaching by delivering a set of modules, and research academics make scholastic impacts through research and publications. Teaching academics prepare and deliver lectures, prepare student assessments, evaluate student performance, and publish grades. Research academics undertake a range of research activities that include conducting research work and publishing research papers. Some academics do both teaching and research. All of these academics are responsible for clarifying student queries and resolving student complaints. They can work for a department on a part time or full time basis.

Every year, the students get enrolled on to the CS courses. From behavioural perspective, they attend lectures for their enrolled modules and appear for assessments to get grades. Students may raise queries in case of any doubt and they may raise complaints for longstanding unanswered concerns/queries.

6.2 Conceptual Model

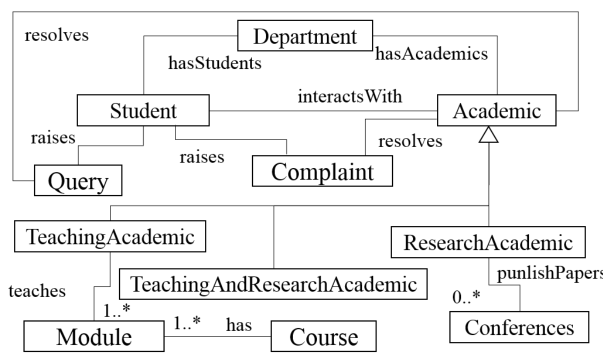


Figure 4: Structural Elements of CS Department

The above problem entity can be described using a set of elements, such as student, academic, classification of the academics, module, courses, conference, query and complaint as illustrated using a class diagram in Figure 4. The key active elements of this structural composition are academics and students. The generic behaviours of the academics and students are depicted using a simple state machine in Figure 5 (a). As shown in the state machine, an individual may start performing an activity from Table 1 for a specific time slot; while performing a started activity the individual moves from `Free` state to `Busy` state, and returns back to `Free` state at the end of it. In addition to this normal behaviour, an individual may suddenly interrupt an activity and returns back to `Free` state to initiate a high priority activity as shown in the state machine.

Precise micro-behaviours of a research academic, teaching academic and student are illustrated using non-deterministic state machines in Figure 5 (b), (c) and (d). As shown in Figure 5 (b), a teaching academic starts preparing for a lecture and eventually reaches the `Prepared` state from `Not Prepared` state. Ideally, the teaching academics should deliver lectures when they are in `Prepared` state. However, they need to deliver lectures based on the department timetable (schedule), which is independent of the state of a teaching academic. Therefore a teaching academic may have to deliver a lecture either from `Prepared` state or from `Not Prepared` state. The teaching academic becomes busy and moves to `Delivering Lecture` state while delivering lecture, and they return to `Prepared` or `Not Prepared` state for the next lecture after delivering a lecture. The non-determinism and temporal uncertainty in the micro-behaviours presented in Figure 5 (b) are: (a) the transition from `Not Prepared` to `Prepared` state, and (b) the transition from `Deliver Lecture` state to `Prepared` or `Not Prepared` state.

The micro-behaviour presented in Figure 5 (c) describes the research and publication behaviour of a research academic. A research academic typically starts with `Inadequate Research` state when they

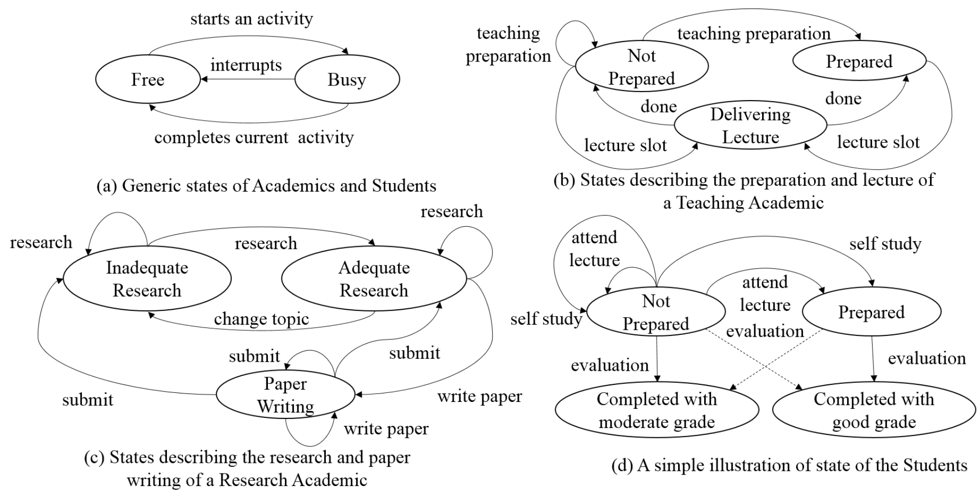


Figure 5: Behaviours of Key Elements CS Department

are new to a research area. Eventually they move to `Adequate Research` state after researching on a topic for a specific period. From `Adequate Research` state, they may continue their research and stay at `Adequate Research` state, return back to `Inadequate Research` state by changing research topic or start writing papers by moving to `Paper Writing` state. From `Paper Writing` state, they have the following options - (i) stay at `Paper Writing` state and continue paper writing, (ii) return back to `Adequate Research` state and continue research work, or (iii) move to `Inadequate Research` when all research ideas are communicated and need further research for new publication.

The micro-behaviour of a student is presented Figure 5 (d). A student attends lectures and does self-study to prepare for modules. The evaluation of a module is conducted based on its schedule; therefore a student may have to appear for an evaluation from any of the two states: `Not Prepared` and `Prepared`. The propensity to reach `Completed with good grade` is high for a student if an evaluation is conducted on `Prepared` state, and `Completed with moderate grade` is high when the evaluation is conducted on `Not Prepared` state. However, there is a probability to reach any of the two states irrespective of their originating state as shown in the figure. The behaviour of the department is not explicitly specifiable as its behaviour emerges from a set interacting micro-behaviours of the academics and students.

6.3 Simulation Model

The conceptual model is specified as an instance of actors meta-model as depicted in Figure 2. An actor based representation of the key elements and enterprise goals are depicted in Figure 6. As shown in the figure. the Academic `Actor` has a set of `Parameters` for characterisation and representing its `State`. For examples, `Parameter` '*workingHours*' decides whether an academic is a full-time/part-time member of staff and '*workPriority*' describes the preference to perform an activity from the list described in Table 1. The `Parameter` '*currentActivity*' captures the state of an academic (*i.e.* 'Free' or 'Busy' with an activity), '*queryRaised*' captures the number of queries raised for the academic, '*complaintReceived*' describes the number of complaints received by the academic and '*workDistribution*' captures the activities performed by an academic (*i.e.*, hourly activities for working days). *Academic* `Actor` receives '*Complaint*' and '*StudentQuery*' `Events` and acts on them by performing '*QueryResolution*' and '*ComplaintResolution*' functional `Behaviours` respectively.

The Academic `Actor` is specialised into three sub-`Actors` to represent teaching academics, research academics and the academics who focus on both research and teaching. The TeachingAcademic `Actor` captures offered modules using '*moduleOffered*' `Parameter` and keeps the records of teaching preparation hours, number of lectures delivered with adequate preparation, number of lectures delivered with less

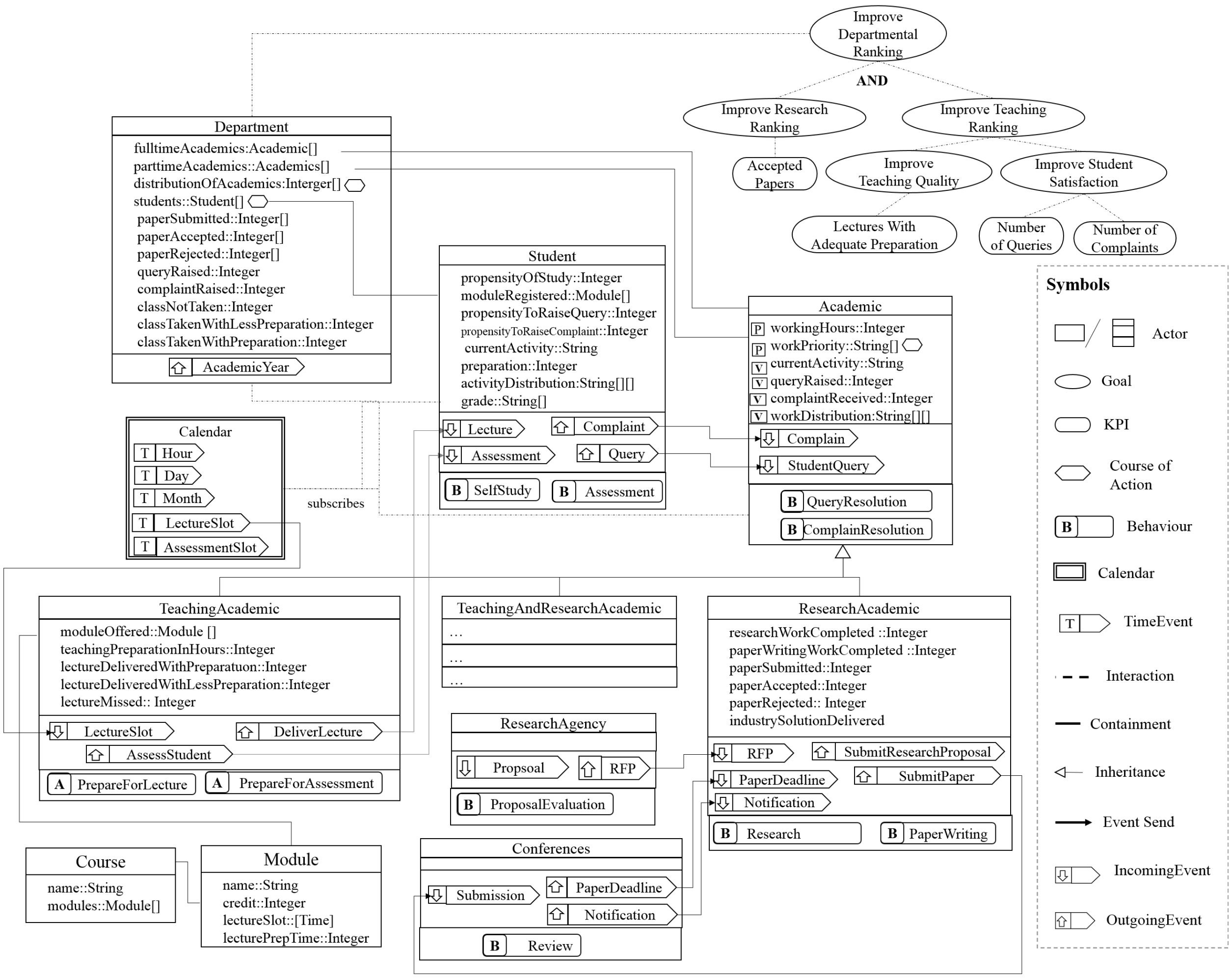
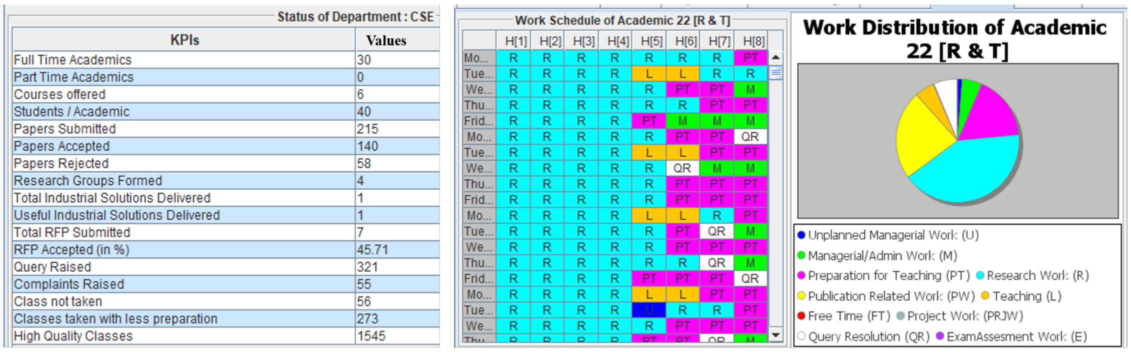


Figure 6: Actor Specification of University Department

preparation and number of lectures missed using '*teachingPreparationInHours*', '*lectureDeliveredWith-Preparation*', '*lectureDeliveredWithLessPreparation*' and '*lectureMissed*' `Parameters`. Behaviorally, it delivers lectures by raising '*DeliverLecture*' `Event` when they receive a '*LectureSlot*' and not busy with high priority activity. A *TeachingAcademic* misses a lecture if the academic is busy with high priority activity. Internally, *TeachingAcademic* prepares for lecture using '*PrepareForLecture*' functional `Behaviour`, which is triggered by an internal an event based on the '*workPriority*' and current state (*i.e.*, '*currentActivity*') of a *TeachingAcademic*. From temporal perspective, the *TeachingAcademic* subscribes `Time Events` specified in `Calendar` as shown in the figure.

A *ResearchAcademic* keeps a record of research done (in hours), time spent on paper writing, number of paper submitted, number of paper accepted and number of paper rejected using '*researchWorkCompleted*', '*paperWritingWorkCompleted*', '*paperSubmitted*', '*paperAccepted*' and '*paperRejected*' `Parameters`. It receives paper deadlines using '*PaperDeadline*' `Event` and may submit papers using '*SubmitPaper*' `Event`. A *ResearchAcademic* researches on a specific topic using '*Research*' `Behaviour` and writes paper using '*WritePaper*' `Behaviour`. They are triggered by the internal events based on the '*workPriority*' and '*currentActivity*' of a *ResearchAcademic*. The `Actor` *TeachingAndResearchAcademic* is the composition of *ResearchAcademic* `Actor` and *TeachingAcademic* `Actor` definitions and their '*workPriority* can be defined based on the activities listed for *TeachingAcademic* and *ResearchAcademic* as shown in Table 1.


Status of Department : CSE

KPIs	Values
Full Time Academics	30
Part Time Academics	0
Courses offered	6
Students / Academic	40
Papers Submitted	215
Papers Accepted	140
Papers Rejected	58
Research Groups Formed	4
Total Industrial Solutions Delivered	1
Useful Industrial Solutions Delivered	1
Total RFP Submitted	7
RFP Accepted (in %)	45.71
Query Raised	321
Complaints Raised	55
Class not taken	56
Classes taken with less preparation	273
High Quality Classes	1545

(a) Simulation dashboard of CS Department (b) Work schedule and status of an academic

Figure 7: Simulation Results

The other core elements, such as *Conference*, *Student* and *Department*, are specified using actors as shown in Figure 6. As depicted, the high-level goal of CS department (*i.e.*, '*Improve Departmental Ranking*'), its sub-goals (*i.e.*, '*Improve Research Ranking*' and '*Improve Teaching Ranking*') and associated KPIs (*e.g.*, '*Accepted Papers*, '*Lectures With Adequate Preparation*', '*Number of Complaints*', and '*complaintRaised*') are specified using actor specification. The specification also defines a `Calendar` that introduces five `Time Events` to represent: 'Hour' (a primitive time event), 'Day' (eight hours), 'Week' (five days), 'Month' (4 weeks), and 'LectureSlot' (a weekly schedule). In addition, the actor specification describes three `Courses of Action` for *what-if* analysis as follows – (a) distribution of the academics (*i.e.*, '*distributionOfAcademic*' – ratio of *TeachingAcademics*, *ResearchAcademics* and *TeachingAndResearchAcademics* of *Department* `Actor`), (b) number of students (*i.e.*, '*students*' `Parameter` of *Department* `Actor`), and (c) work priority of academics (*i.e.*, '*workPriority*' `Parameter` of *Academic* `Actor`).

6.4 Instantiation, Simulation and Decision Making

For experimentation, the above actor model is instantiated to represent a specific *Department*. In this case, the '*CS Department*' is formed using 1200 *Students* and 30 full time *Academics* with 20:60:20 percent distribution of *TeachingAcademics*, *TeachingAndResearchAcademics* and *ResearchAcademics* with varying work priorities. The '*CS Department*' is configured to offer six *Courses* where each *Course* is comprised of four *Modules*. In this formulation, the instances of *TeachingAcademic* offer one or more *Modules* and they consider unplanned activities, such as addressing queries and complaint, as a high priority activity. The rest of the activities are prioritised as: teaching work, preparation for teaching and then assessment. On the other hand, the instances of *ResearchAcademic* prioritise their activities as: complaint resolution, research work, writing paper, other unplanned activities.

The configured and instantiated actor model is specified using ESL,simulated for one '*Year*' using ESL engine and `KPIs` are observed. The simulation dashboard showing an overview of the departmental `KPIs` are shown in Figure 7 (a). The Dashboard shows – 215 papers are submitted in a '*Year*', 140 papers got accepted, 58 paper got rejected (rest of the papers are in review process), 321 queries are raised by the students, 55 complaints are raised by the students, a total of 55 lectures are missed by the academics, 273 lectures are delivered with less preparation and 1545 lectures are delivered with adequate preparation.

The trace and latest snapshot of the work distribution of an academic are respectively shown using a table and a pie-chart in Figure 7 (b). The graphs of all academics and consolidated results show an indicative state of CS department for the stated configuration. An operation validation can be performed at this stage by configuring a known scenario and comparing observed results. A validated model, *i.e.*, faithful virtual representation of a real enterprise, can then be used for *what-if* analyses.

In this case, the decision questions are what will be state of CS department (after one year) if it adopts different configurations, such as more research academics, more teaching academics, different distributions of research and teaching academics, different priorities of the academics or different students/academics

	Courses of action	Full Time Academics	Students / Academic	Papers Submitted	Papers Accepted	Papers Rejected	Query Raised	Complains Raised	Class not taken	Classes taken with less preparation	Classes taken with adequate preparation
1	Initial Configuration<T = 20, R&T= 60, R = 20>	30	40	215	140	58	321	55	56	273	1545
2	Focus on Research <T = 20% , R&T=40%, R= 40%>	30	40	228	145	65	323	50	56	318	1527
3	Preference to Teaching Work for Teaching Academics.	30	40	224	114	90	212	62	64	153	1693
4	More Teaching Staff <T = 30% , R&T=50%, R= 20%>	30	40	185	124	43	181	45	49	178	1684
5	Improving Academics per Student Ratio	40	30	164	91	69	255	46	50	210	1632
6	Experiment with distribution <T = 35% , R&T=35%, R= 30%>	30	40	246	156	62	157	51	58	107	1719

Figure 8: Consolidated Simulation Results

ratio. These scenarios are explored by changing department composition and observing simulation results. The observations are captured using a table as shown in Figure 8. The observations are: adding more research academics may not lead to better research outcomes (row 2). The change in work priority of the teaching academics (*i.e.* prioritising teaching activity as compared to query resolution and complain resolution) helps in improving teaching quality but negatively impacts the students satisfaction (row 3). Increase of teaching academics improves the student satisfaction but other goals remain unaddressed (row 4). A better student/academic ratio (row 5) may not be an useful course of action unless the additional academics are recruited appropriately as shown in Row 5. A distribution of teaching, research and teaching, research academics = $<35, 35, 30>$, as shown in row 6, produces most desirable outcome among the alternative experimented in this case study. This experiment provides an indication for further explorations. For instance, the table shown in Figure 8 indicates a possibility to arrive at a better alternative by combining better distribution (as shown in row 6) and better work preference (as shown in row 3). Multiple such explorations help decision makers to arrive at a decision, which is backed by quantitative evidences. Moreover, these simulation explorations provide an indication of the positive and negative consequences of the alternatives before implementing them in reality.

7 CONCLUDING REMARKS

The research described in this paper has presented an approach to quantitatively evaluate possible control and adaptation considerations using an actor-based simulation-driven digital twin. An effective adoption of the method recommended by Robert Sargent for model creation and model validation within an actor-based modelling paradigm is proposed and validated. The key observations are – (a) the digital twin based exploration is cost and time effective approach for analysing business enterprises, (b) the notion of actor is pragmatic consideration for representing complex enterprises, and (c) the augmentation of canonical simulation method and validation techniques to actor-based digital twin improves the faithfulness of the constructed digital twin. Overall, they enable a quantitative, evidence-driven, informed decision-making of complex enterprises. As the next steps, further explorations are beneficial for improving the utility of the proposed approach – (a) the construction of digital twin requires significant involvement of domain experts – a visual domain specific language for actor model could be a way forward, (b) establishing the faithfulness of a digital twin requires multiple iterations – supplementing model validation with behavioural type checking (Gay and Ravara 2017) could be useful, and (c) introduction of reinforcement learning agent as decision-making unit is another useful aid for decision-makers.

REFERENCES

Agha, G. 1986. *Actors: A Model of Concurrent Computation in Distributed Systems*. Cambridge, MA, USA: MIT Press.

Allen, J. 2013. *Effective Akka*. O'Reilly Media, Inc.

Altbach, P. 2007. "The Dilemmas of Ranking in: International Higher Education". *Journal of Studies in International Education* 3(6):121–133.

Armstrong, J. 1996. "Erlang - a Survey of the Language and its Industrial Applications". In *In Proceedings of the symposium on industrial applications of Prolog (INAP)*, 8.

Boschert, S., and R. Rosen. 2016. "Digital twin – the simulation aspect". In *Mechatronic Futures*, 59–74. Springer.

Camus, B., C. Bourjot, and V. Chevrier. 2015. "Combining DEVS with multi-agent concepts to design and simulate multi-models of complex systems (WIP)". In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 85–90.

Candes, E. J., and T. Tao. 2005. "Decoding by linear programming". *IEEE transactions on information theory* 51(12):4203–4215.

Casti, J. L. 1994. *Complexification explaining a paradoxical world through the science of surprise*. HarperPerennial - A Division of Harper Collins Publishers.

Clark, T., V. Kulkarni, S. Barat, and B. Barn. 2017. "ESL: An Actor-Based Platform for Developing Emergent Behaviour Organisation Simulations". In *Practical Applications of Agents and Multi-Agent Systems*, 311–315. Springer.

Daft, R. L., and A. Y. Lewin. 1990. "Can organization studies begin to break out of the normal science straitjacket? An editorial essay". *Organization Science* 1(1):1–9.

Forrester, J. W. 1994. "System dynamics, systems thinking, and soft OR". *System dynamics review* 10(2-3):245–256.

Gay, S., and A. Ravara. 2017. *Behavioural Types: from Theory to Tools*. River Publishers.

Glaessgen, E., and D. Stargel. 2012. "The digital twin paradigm for future NASA and US Air Force vehicles". In *Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 1818.

Grieves, M., and J. Vickers. 2017. "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems". In *Transdisciplinary perspectives on complex systems*, 85–113. Springer.

Hewitt, C. 2010. "Actor model of computation: scalable robust information systems". *arXiv preprint arXiv:1008.1459*.

Holland, J. H. 2006. "Studying complex adaptive systems". *Journal of Systems Science and Complexity* 19(1):1–8.

Iacob, M., D. H. Jonkers, M. Lankhorst, E. Proper, and D. D. Quartel. 2012. "ArchiMate 2.0 Specification: The Open Group". Van Haren Publishing.

Lukman, R., D. Krajnc, and P. Glavič. 2010. "University ranking using research, educational and environmental indicators". *Journal of Cleaner Production* 18(7):619–628.

Macal, C. M., and M. J. North. 2010. "Tutorial on agent-based modelling and simulation". *Journal of simulation* 4(3):151–162.

Sargent, R. G. 2005. "Verification and validation of simulation models". In *Proceedings of the 37th conference on Winter simulation*, 130–143.

Schleich, B., N. Anwer, L. Mathieu, and S. Wartzack. 2017. "Shaping the digital twin for design and production engineering". *CIRP Annals* 66(1):141–144.

Simon, H. A. 1991. "The architecture of complexity". In *Facets of systems science*, 457–476. Springer.

Sutton, R. S., A. G. Barto et al. 1998. *Introduction to reinforcement learning*, Volume 135. MIT press Cambridge.

Tolk, A., B. L. Heath, M. Ihrig, J. J. Padilla, E. H. Page, E. D. Suarez, C. Szabo, P. Weirich, and L. Yilmaz. 2013. "Epistemology of modeling and simulation". In *Proceedings of the 2013 Winter Simulation Conference*, 1152–1166. IEEE Press.

Von Bertalanffy, L. 1968. "General system theory". *New York* 41973(1968):40.

White, S. A. 2008. *BPMN modeling and reference guide: understanding and using BPMN*. Future Strategies Inc.

Zachman, J. A. 1987. "A framework for information systems architecture". *IBM systems journal* 26(3):276–292.

AUTHOR BIOGRAPHIES

SOUVIK BARAT is Senior Scientist at Tata Consultancy Services Research (TCSR). He holds a PhD from Middlesex University London. His research interests include simulation, agent and actor technology, enterprise modeling and digital twin. His email address is souvik.barat@tcs.com.

VINAY KULKARNI is Chief Scientist at Tata Consultancy Services Research (TCSR). His research interests include model-driven software engineering, self-adaptive systems, and enterprise modeling. Vinay also serves as a Visiting Professor at Middlesex University London. His email address is vinay.vkulkarni@tcs.com.

TONY CLARK is Professor of Software Engineering at Aston University in the UK. His academic research on meta-modelling led to the development of a tool called XModeler that has been used in a number of commercial applications including the development of tool support for a new Enterprise Architecture modeling language. His email address is tony.clark@aston.ac.uk.

BALBIR BARN is Professor of Software Engineering and Deputy Dean at Middlesex University's School of Engineering and Information Science. His academic research interests include model driven organisation, conceptual modeling, enterprise modeling, epistemology of simulation. His email address is b.barn@mdx.ac.uk.