

TOWARDS ADAPTIVE ENTERPRISES USING DIGITAL TWINS

Vinay Kulkarni
Souvik Barat

TCS Research
54B, Industrial Estate
Hadapsar, Pune
411013, INDIA

Tony Clark

School of Engineering and Applied Sciences
Aston University
Aston St, Birmingham
B4 7ET, UK

ABSTRACT

Modern enterprises are large complex systems operating in highly dynamic environments thus requiring quick response to a variety of change drivers. Moreover, they are systems of systems wherein understanding is available in localized contexts only and that too is typically partial and uncertain. With the overall system behaviour hard to know a-priori and conventional techniques for system-wide analysis either lacking in rigour or defeated by the scale of the problem, the current practice often exclusively relies on human expertise for monitoring and adaptation. We present an approach that combines ideas from modeling & simulation, reinforcement learning and control theory to make enterprises adaptive. The approach hinges on the concept of Digital Twin - a set of relevant models that are amenable to analysis and simulation. The paper describes illustration of approach in two real world use cases.

1 INTRODUCTION

Modern enterprises are complex systems of systems operating in highly dynamic environments that need to respond quickly to a variety of change drivers. Determining the right response often requires a deep understanding of aspects such as structural decomposition into subsystems, relationships between these subsystems, and emergent behaviour. The scale of organisations, their socio-technical characteristics, and fast business dynamics make this a challenging endeavour. Current industry practice relies principally on human expertise to arrive at suitable responses and has turned out to be inadequate. Several enterprise modelling languages have been proposed to aid human-in-the-loop decision making, however, none fully addresses the problem. Languages capable of specifying all relevant aspects are not amenable to rigorous analysis, and the languages that provide sophisticated analysis capabilities address only a subset of the relevant aspects. Moreover, lack of interoperability makes it a challenge to use a set of relevant languages together. Furthermore, implementing the desired response through suitable modifications to enterprise IT systems, business processes and strategies is no less complex. Little help exists to identify what needs to be modified where and how.

Having identified the desired set of modifications, technology exists to introduce these changes to an enterprise, albeit in a widely varying spectrum of efficacy. For instance, modern IT systems are relatively more amenable to such modifications than legacy systems. However, adapting enterprises suitably and effectively in increasingly dynamic environments remains a time- and effort -intensive endeavour.

To address the problem of enterprise adaptation, in this tutorial, we present a novel approach that is based on domain knowledge, is driven by real-world data, and that uses simulation models to support evidence based adaptation. The approach hinges on the concept of Digital Twin – a set of relevant models amenable to rigorous analysis and simulation. We describe the core modeling and model processing

infrastructure necessary to support the proposed approach. We describe early stage explorations of its application to problems where the mechanistic world view holds. We argue similar benefits are possible for problem spaces involving human actors as well.

2 STATE OF THE ART AND PRACTICE

The analysis of complex enterprises is typically approached using two broad categories: qualitative approach (McMillan, 1980) and quantitative approach (Currall and Towler, 2003). The qualitative approach is concerned with the subjective assessment of the underlying enterprise through a range of management techniques such as interviews, discussions, and field studies. The quantitative approach, in contrast, involves precise interpretation of system data, structure and behaviours.

The quantitative approach is further classified into three categories: (a) *inferential approach*, (b) *experimental approach* and (c) *modelling and simulation approach* (Kothari, 2004). The inferential approach (Michalski, 1993) analyses the existing system data (*i.e.*, trace or historical data) to infer the characteristics of a system or an enterprise. This approach is effective when the environment where an enterprise operates is static. The experimental approach comprehends an enterprise by manipulating the system variables and observing their effects in a controlled environment. This approach is often infeasible or not an economical option for large business critical enterprises.

The modelling and simulation approach imitates a real enterprise using a (purposive) model, explores a range of scenarios by simulating the possible (forward looking) changes incorporated into the model, and develops a precise understanding about an enterprise by interpreting the simulation results. The modelling and simulation approach visualises systems using two broad approaches: top-down approach and bottom-up approach (Thomas and McGarry, 1994). A top-down approach models an enterprise as a whole and adopts reductionist view to decompose it into smaller parts to understand the parts in isolation. This approach uses a range of models to represent and analyse enterprises. These models are: (i) mathematic model and (ii) enterprise model (EM). The mathematical models, such as linear programming (Candes and Tao, 2005) and integer programming (Schrijver, 1998), represent a system using mathematical formulae and use rigorous mathematical and statistical problem solving techniques for system analysis. The enterprise models (EMs), such as ArchiMate (Iacob, et al. 2012), *i** (Yu et al. 2006) and BPMN (White, 2008), and System Dynamics (SD) (Meadows and Wright, 2008), are typically less rigorous than mathematical models, however they serve a wide range of modelling and analysis needs of the complex enterprises. The key concerns with top-down approaches are: they are not cognizant of emergent behaviour and expect information about whole enterprise, which is a difficult expectation for a large enterprise.

A bottom-up approach starts from the parts or micro-behaviours and arrives at a holistic view of a system through composition. The bottom up approach uses the agent and actor based technologies, such as Erlang (Armstrong, 1996), Akka (Allen, 2013), and Scala Actor (Haller and Odersky, 2009), for modelling and analysing systems. They are capable of observing emergent behaviour but not capable of representing complex structure and not cognizant of uncertainties.

3 PROPOSED SOLUTION

We illustrate the knowledge-guided data-driven model-based simulation-aided evidence-backed approach for design, control and adaptation of large enterprises. Though validated in problem spaces where the mechanistic world view holds, we argue, it is equally applicable for socio-techno spaces as well. Though not described here for want of space, the tutorial will outline a line of attack for this aspect as well.

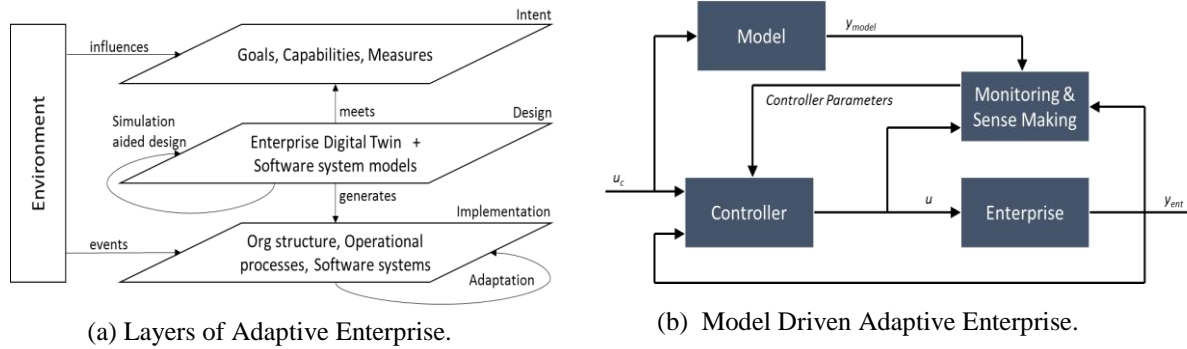


Figure 1: Adaptive Enterprise.

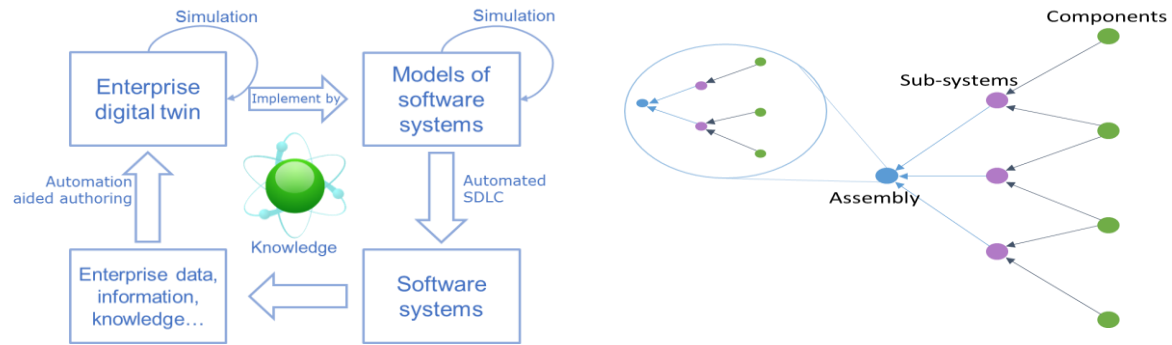
3.1 Line of Attack

We envisage a line of attack wherein an enterprise is viewed at three related planes namely: *Intent* defining goals, capabilities, and measures; *Design* defining organisational structure, processes, and information systems; *Implementation* realizing the design plane in terms of organisation and its software systems; and interactions of the three planes with an environment as shown in Figure 1 (a). The design plane needs to provide suitable machinery to arrive at the specifications of the right organisational structure, processes and information systems so as to meet the specified intent. To this end, we propose simulation-guided data-driven machinery for design space exploration. The outcome of this human-in-loop process is an executable model of the enterprise and specifications of its software systems – *i.e.* an enterprise *digital twin*. We use model driven techniques to transform the design plane specifications into an efficient implementation of supporting software systems that are capable of dynamically adapting to the changes in environment. The design plane models serve as the reference for adaptation of implementation plane systems.

Our adaptation architecture draws its inspiration from a well-studied and widely used concept from control theory – Model Reference Adaptive Control (Butler et al. 1992) shown in Figure 1 (b). The Model captures the desired reference behaviour of enterprise. The Enterprise is the complex system of systems to be controlled - viewed principally as input-output transfer function and possibly persistent side-effects. The Monitoring & Sense Making component constitutes the core technology infrastructure to observe and discern input and output of enterprise. The Controller component constitutes the core technology infrastructure that, together with the Monitoring & Sense Making component, nudges the Enterprise as close to the Model as possible thus achieving a model-guided data-driven justification-based human-in-the-loop adaptive response.

In manufacturing domain terminology, the fixed point of the design plane model is a digital twin of the enterprise. A digital twin mimics the behaviour of a system in order to support what-if analyses and to arrive at appropriate responses to various contingencies that may arise in a plant.

Future enterprises are systems of systems with complex interactions operating in a dynamic environment. Given the structural and behavioral complexity, detailed understanding is possible only in localized contexts (Bar-Yam 1997). At the same time, events occurring in one context influence the outcomes in other contexts. Lack of complete information coupled with inherent uncertainty make holistic analysis of systems intractable. As a result, decisions pertaining to system design and implementation are unlikely to be globally optimal. Non-availability of complete information and inherent uncertainty make traditional optimization approaches impractical. Therefore, in many cases a simulation-based approach is the only recourse available for arriving at a “good enough” solution by navigating the solution space (Gosavi 2003). However, considering the open nature of the problem, an exhaustive navigation of the solution space is infeasible. It calls for intelligent navigation of the solution space guided by domain knowledge and learning from past experience. Figure 2(a) provides a pictorial representation of proposed line of attack that hinges on: (i) model-based machinery to help define enterprise digital twin, (ii) simulation



(a) Model-based Architecture for Adaptive Enterprise. (b) Automobile Manufacturing System of Systems.

Figure 2: Architecture for Adaptive Enterprise.

machinery what-if and if-what analysis, (iii) mode I-based machinery to help implement the system right, (iv) a mechanism to map the two sets of models along with a means to derive one from the other, and (v) a learning loop to enable continuous improvement over time.

3.2 Complex System of Systems

Consider a large automobile manufacturing company. It is essentially a geographically distributed ecosystem comprising of component manufacturers, sub-system manufacturers, and an assembly plant. The assembly plant itself is an ecosystem comprising of chassis manufacturing unit, body fabrication unit, paint shop, and a set of assembly lines. This complex system of systems is serviced downstream by another system of systems comprising of warehouses, dealerships, service centers etc. Figure 2 (b) depicts the automobile manufacturing company as a hierarchically decomposable system of systems. Steady state operation of this system is typically governed by production schedule arrived at centrally based on the demand, supply and capacity considerations. Suppose there is a disturbance at a few of these component manufacturers. While it is possible to know with a good degree of precision the impact of this disturbance on production schedules of these manufacturers, the downstream ramifications of this disturbance such as impact on production schedules of sub-system manufacturers and assembly plant, inventory, order cancellations *etc.* and in turn the effect of these events on the upstream supply chain are difficult to assess and quantify. This is largely due to complex interactions among the various constituents of ecosystem and consequent inability to predict the global impact of local actions. Typically, even in a localized context, while one broadly knows what corrective actions to perform for what problems and potential outcomes of those actions, one does not always know what the precise problem is and what the precise outcomes of the various mitigating actions are. This local uncertainty gets further exacerbated as it is propagated through the ecosystem. The dynamic nature of ecosystem *i.e.*, specific constituents and their behaviours changing over time, makes the problem even harder. The key problems can be summarized as: (i) How to understand emergent behavior of complex system of systems (*i.e.*, analysis), (ii) How to continue delivering the stated objectives in presence of perturbations (*i.e.*, control), (iii) How to survive threats and maximize opportunities (*i.e.*, transformation), and (iv) How to provide a feel for the system with a priori assurances (*i.e.*, design).

3.3 Enterprise Digital Twin

In engineering discipline, modeling is the means to address problems of the kind discussed above. Models are precise specifications of the essential aspects of system. These specifications are amenable to quantitative analysis through what-if and if-what scenario playing. Such models are called a Digital Twin of the real system. Similar digital twins are required for enterprises as well. An enterprise digital twin is a

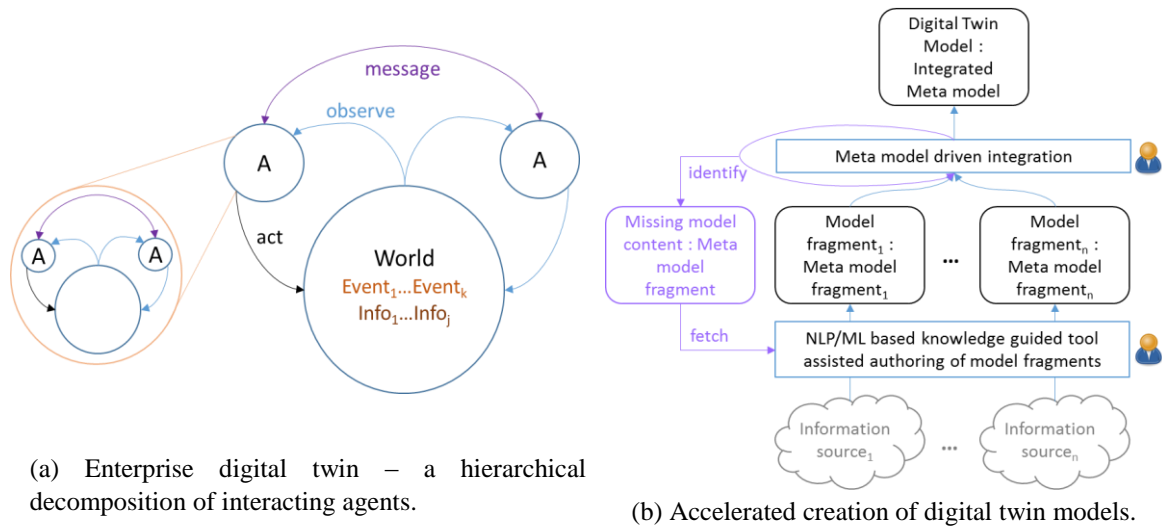


Figure 3: Digital Twin Models.

virtual, high fidelity representation of complex system of systems that is amenable to rigorous quantitative analysis through what-if and if-what scenario playing using real data to facilitate local optimality, global robustness and continuous learning. An enterprise digital twin models an enterprise as a set of intentional autonomous adaptive learning agents that interact with each other and respond to the events of interest taking place in the environment. Agents can exist at different levels of granularity *i.e.*, an agent can be seen as a composition of a set of next-level agents as shown in Figure 3 (a). An agent observes the environment, makes sense of the observations, and performs actions so as to achieve its objectives. The action could be changing the local state of agent or sending a message to other agents. These actions can be stochastic to model uncertainty. An agent is capable of adapting its behavior in response to the changes in its environment. Essentially, an agent has a set of situation-specific behaviours and it is able to switch from one behavior to another depending on the situation it finds itself in. An agent adapts its behaviour not only to achieve its objectives but also to ensure robustness of the overall system. An agent is also capable of learning new behaviours.

An enterprise digital twin is a set of purposive analyzable and simulatable models representing the enterprise in order to mimic real-world phenomenon. A variety of Enterprise Modeling (EM) languages exist that provide information-capture and analysis support across a wide spectrum of sophistication. The majority of these languages can be traced to Zachman framework (Zachman 1987) advocating that capture of the why, what, how, who, when and where aspects leads to the necessary and sufficient information for addressing a given problem. Thus it can be argued that a complete specification of an enterprise is possible using the Zachman framework, however, there exists no support for analysis as the information is captured typically in the form of texts and pictures. It can be observed of the existing Enterprise Modelling (EM) languages that: the languages capable of specifying all the relevant aspects of an enterprise for organisational decision-making lack support for automated analysis (Zachman 1987, Krogstie 2008, Jonkers et al. 2004), and the languages capable of automated analysis can only address a subset of the aspects required for decision-making (Yu et al. 2006, Meadows and Wright 2008, White 2004). Moreover, the system of systems nature and the large size of modern enterprises means that an understanding of an enterprise – structural and behavioural – is available only locally, from which the overall behaviour needs to be derived. Also, even the local understanding can have an element of uncertainty.

The Actor model is based on the model of concurrent computation wherein an *Actor*, in response to a message that it receives, can: make local decisions, create more actors, send more messages, and determine how to respond to the next message received. Actors may modify their own private state, but can only affect

each other through messages avoiding the need for any locks (Hewitt et al. 1973). We have developed an Actor-based modelling language, ESL (Clark et al. 2017a), to specify an enterprise as a set of autonomous encapsulating units that interact with each other by exchanging messages (Clark et al. 2017a). We have developed a component abstraction to model the fractal nature of system of systems (Kulkarni et al. 2014). We have extended the actor-based paradigm to enable modelling of uncertainty (Kulkarni et al. 2017). Through a set of case studies, we have validated the adequacy of ESL to specify complex system of systems for decision-making (Barat et al. 2017a).

3.4 Creating Digital Twin Models

Creating the right models at the right level of abstraction is the key to efficacy. These models are typically large with the required information spread across the breadth of enterprise in various forms such as data bases, execution logs, standard operational procedure notes, policy documents and so on. As a result, an army of experts from wide-ranging fields of expertise is required to manually create the digital twin models using appropriate model editors – clearly a time-, cost-, and effort-intensive endeavour. Figure 3 (b) gives a pictorial overview of the framework we have developed for accelerated creation of digital twin models from information available in semi- / un-/ structured form. It comprises of automation aids based on: (i) Natural Language Processing (NLP) and Machine Learning (ML) techniques for gathering the desired information from a given information source (ii) meta model driven techniques for integration and reconciliation of the model fragments, and (iii) model validation based techniques for identifying the missing model fragments. However, quality of the models created is largely dependent on the knowledge content of the domain expert using these aids.

We provide two ways of validating the digital twin models: (i) certification of correctness by experts, and (ii) through simulation wherein the models (suitably initialized) are subjected to known past events leading to simulation trace which is then examined to ascertain the results are identical to the ones from the past. Our simulation engine generates rich execution traces containing detailed information necessary for analysis. We have developed a pattern language to specify the desired behaviour and a pattern matching engine to look for the specified patterns in the simulation trace (Clark et al. 2017b). This generic solution to ascertain correctness can be further augmented by manual validation of the input, output and control variables of the simulation. This, we believe, should cover a wide range of digital twin models.

3.5 Using Digital Twin

Enterprise digital twin can help with analysis, control, transformation and design of enterprise as a complex system of systems.

3.5.1 Analysis

Enterprise is a complex system of systems operating in a dynamic environment. As a result, it is subject to a variety of changes taking place in its environment. It is essential to understand how the system is responding to these changes and the emergent behavior. Traditionally, this has been the preserve of experts who rely on their knowledge to analyze the available data to come up with remedial changes. Large size, complex interactions and high dynamics make this a highly difficult task to be performed manually. At best, this can be done when limited only to localized contexts. However, there is no guarantee that remedial actions thus arrived at can ensure global correctness let alone optimality.

On the other hand, being a virtual high fidelity simulatable representation of the real system, enterprise digital twin enables what-if scenario playing in virtual space. Thus it is possible to observe both local and emergent global behaviours virtually thus giving an accurate understanding of what is happening in real system. Enterprise digital twin also enables if-what analysis, for instance, whether a desired global behavior can be achieved and how.

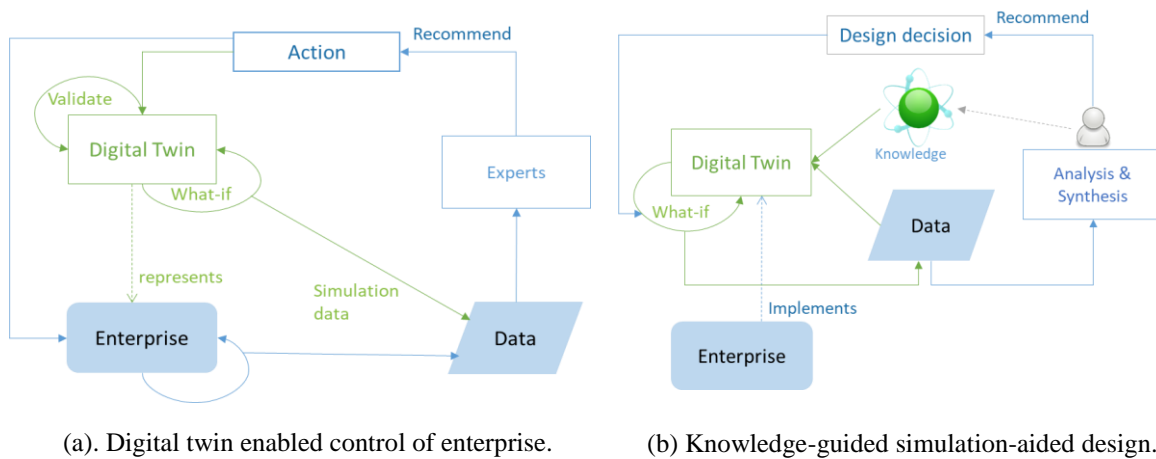


Figure 4: Control and Design using Digital Twin.

3.5.2 Adaptive Control

Enterprises need to deliver the intended goals in presence of internal and external perturbations. With increased dynamism the frequency of such perturbations will only increase. Traditionally, these perturbations are handled by experts looking at past data to come up with recommendations as shown in Figure 4 (a). However, this approach suffers from a disadvantage that recommendations are arrived at based only on situations that are experienced so far. These recommendations are effected on real systems and monitored for their effectiveness. This has a certain trial-and-error element to it leading to long cycle times and high costs with effectiveness of recommendations depending largely on the level of expertise. In contrast, enterprise digital twin improves effectiveness by enabling validation of recommendations in virtual space. Also, enterprise digital twin can be subjected to a variety of scenarios a priori thus obtaining a more comprehensive data pertaining to situations that are possible though not seen so far.

Enterprise digital twin can also be used for self-adaptation thus reducing dependence on human experts. Figure 1(b) depicts an adaptation architecture that draws its inspiration from a well-studied and widely used concept from control theory – Model Reference Adaptive Control (MRAC) – adapted for enterprise context. The Model captures the desired reference behaviour of enterprise. The Enterprise is the complex system of systems to be controlled. The Monitoring & Sense Making constitutes the core technology infrastructure to observe and discern input and output of enterprise. The Controller constitutes the core technology infrastructure that, together with the Monitoring & Sense Making component, nudges the Enterprise as close to the Model as possible thus achieving a model-guided adaptive response. For the digital enterprise, Enterprise Digital Twin serves as the Model which is fed the same input as Enterprise thus producing a reference output. The Monitoring & Sense making component compares the Enterprise output with the reference output to identify the distance if any. The Controller is guided by the input, the Enterprise output and the distance signal to nudge the Enterprise behavior in the right direction.

3.5.3 Design

Design is the process to arrive at the right strategies, processes, organizational structure and supporting systems to achieve the intended goals and objectives. Traditionally, it is a knowledge-intensive endeavor wherein experts analyze and synthesize available data to come up with design decisions. These decisions are typically implemented using a set of tried-and-tested templates. The resultant implementation may then be validated using controlled experimentation where the experts assess whether the design is meeting the intended goals. This process has long cycle times especially since each controlled experiment may involve

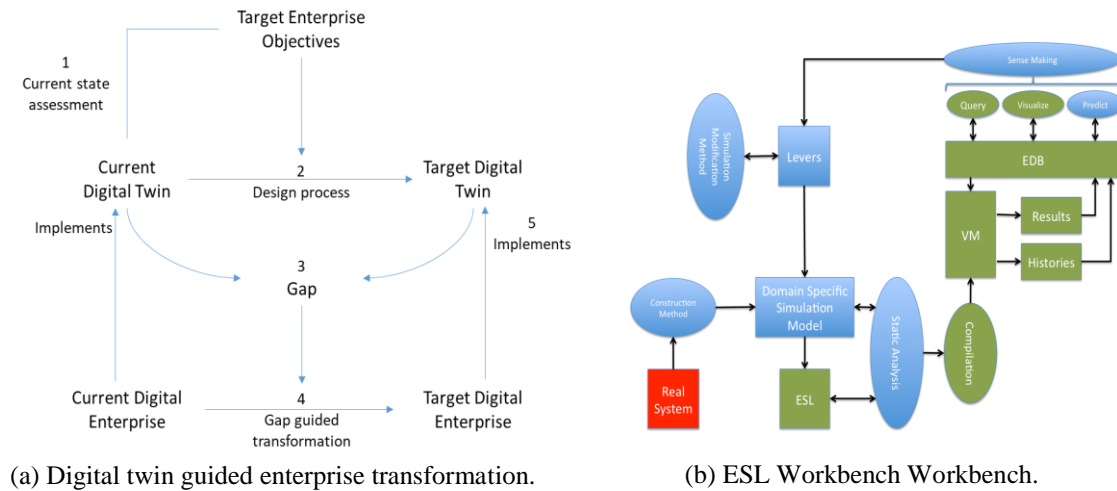


Figure 5: Digital Twin based transformation and enabling technology.

setting up the necessary physical infrastructure, assigning resources, and conducting the experiments in realistic conditions. As a result, this approach is quite expensive too.

In contrast, Enterprise Digital Twin can help explore the design space virtually as shown in Figure 4 (b). Domain knowledge and past data help come up with first-cut definitions of goals, strategies, policies, processes, organization structure *etc.* which are then modelled into the Enterprise Digital Twin. The simulatable nature of Enterprise Digital Twin enables what-if and if-what analysis of various scenarios of interest to arrive at the right design decisions virtually. These decisions can then be validated using the digital twin itself. This iterative process goes on until the desired objectives are met. Implementation of the real system is a realization of the digital twin. As all the artefacts are models and solution space exploration is ‘in silico’, the design process is much faster and cheaper. Model-based analysis significantly reduces dependence on human expertise. Results of simulation can be mined to continuously augment the domain knowledge.

3.5.4 Transformation

Enterprises face disruptions due to emergence of new technologies, new business models, new regulatory frameworks, business events such as mergers and acquisitions etc. These disruptions necessitate large scale changes in enterprise strategies, organizational structure, processes, systems etc. Due to the interconnected nature of complex systems, these changes ripple across. As the interactions are complex, it is difficult to predict the extent and exact nature of these ripple effects. Current practice of relying largely on human experts for effecting such transformations has clearly been found wanting. By some estimates, only 3% of enterprise transformation projects ever come to completion¹.

Enterprise Digital Twin enables a transformation architecture that provides a tractable path to arrive at the desired target state as shown in Figure 5 (a). The transformation process begins by specifying the target state. The digital twin of current enterprise can be used provide an accurate assessment whether the target objectives can be achieved by the current enterprise. This can be supported through what-if and if-what analysis on the digital twin of current enterprise. In case current enterprise cannot be made to achieve the desired target objectives, it needs to be structurally and behaviourally transformed. Transformation process then identifies digital twin for the desired target enterprise – this is essentially the design process described earlier starting with the current digital twin. It is followed by identifying the gap between current and target digital twins in terms of the various models. These gaps drive the transformation of current digital enterprise

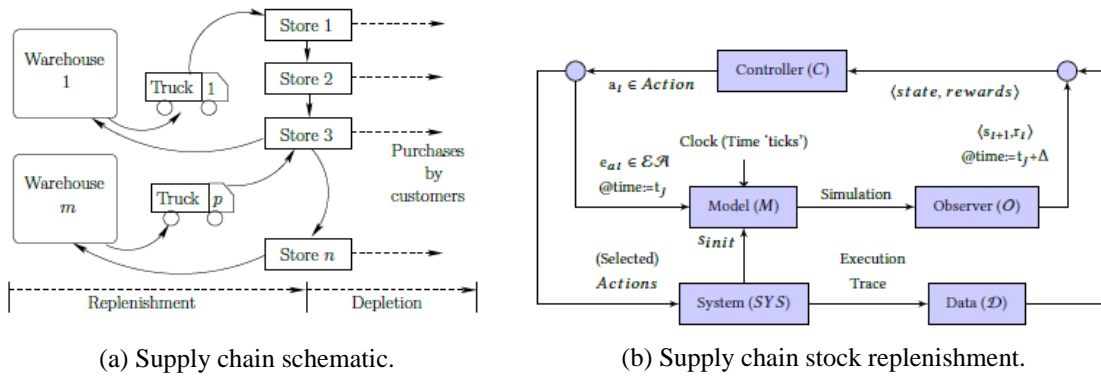


Figure 6: Overview of Supply Chain Case Study.

into the target digital enterprise. Final step of the transformation process checks whether the target digital enterprise correctly implements the target digital twin.

3.6 The ESL Workbench

The ESL workbench supports an Actor-based language for simulation-based decision-making (Clark et al. 2017a). The workbench is shown in Figure 5 (b) and will be introduced as part of the tutorial.

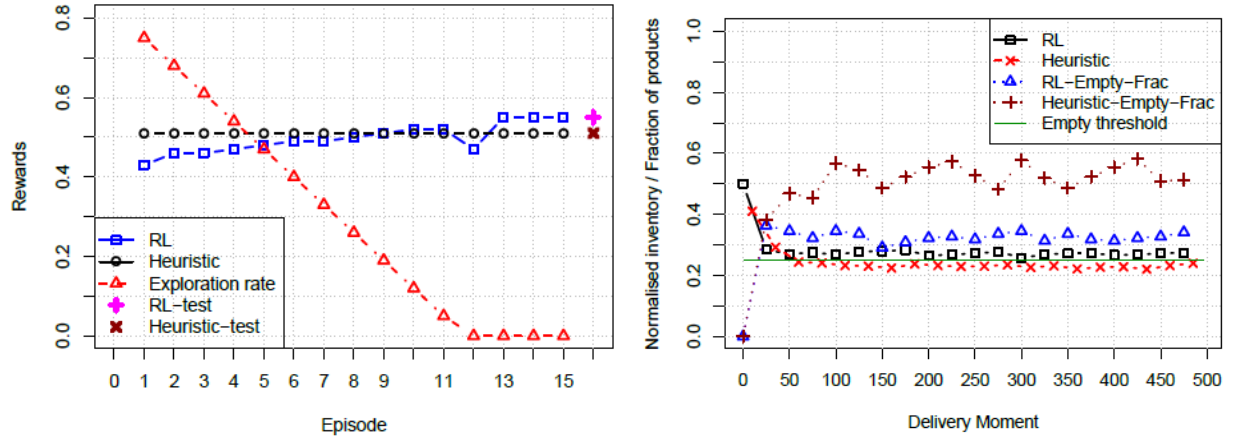
ESL is an actor language that has been designed to run applications and simulations that require information-rich processing by autonomous computational units. Each actor runs in its own thread of control and communicates with other actors using asynchronous messages. ESL supports pattern-matching, data locks and higher-order functions. Applications in ESL are highly concurrent and are driven by system generated time-events. ESL can support sense-making of applications and simulations by generating state histories that record computation steps; once generated, a history can be interrogated using a temporal-logic based query language. ESL has a static type system that includes parametric polymorphism.

Each actor has a behaviour that can be modelled as a state machine. When implemented in ESL the model can be run to produce emergent behaviour that is subsequently analysed through the EDB component of the ESL Workbench. This includes visualisation of the simulation results and analysis of the simulation history (Clark et al. 2017b). The shop owner can speculate about the distribution of customers with particular types of behaviour and run the simulation multiple times in order to determine the outcome. The tutorial will use a number of examples similar to the shop simulation in order to introduce the participants to the ESL Workbench.

ESL application development is supported by a tool called EDB that provides real-time support for syntax and type checking. EDB can be used to create and manage an ESL application including its execution, debugging and history management. EDB allows ESL to generate graphical representations of running applications and of their histories.

3.7 Model Driven Techniques for Adaptive Software Systems

Modelling helps address software development and integration from a higher level of abstraction. Most prominent benefit delivered so far has been through automated code generation (Kulkarni 2016, John et al. 2011). The software system is specified in terms of the what and how aspects leaving out the details such as architectural decisions, design strategies, and implementation technology platform. A set of appropriate code generators transform these specifications into a platform specific implementation while adding details pertaining to the chosen architectural decisions and design strategies. The same specification can be used to deliver another solution with different choices for architectural decisions, design strategies, and implementation technology platform. Support for variability modelling & resolution at model level enables product lines (Kulkarni 2010, Kulkarni and Barat 2011). Model based code generators also benefit from



(a) Digital Twin enabled RL-based controller outperforms aggregated heuristics while balancing explore Vs exploit. (b) Digital Twin enabled RL-based controller outperforms heuristics as regards inventory.

Figure 7: Experimental Results

model based code generation thus further improving productivity and agility (Kulkarni and Reddy 2008). Several manifestations of this technology exist (TCS 2019), however, the focus so far has been on delivering a bespoke solution or at best a product line *i.e.*, addressing largely static scenarios.

Software systems should reflect the characteristics of the enterprises they support. As enterprises are evolving into complex system of systems that need to quickly adapt to a variety of changes taking place in their operating environment, so should their supporting software systems. This puts new requirement of adaptiveness on enterprise software systems to be achieved through sense-and-respond architectures.

Though MAPE-K architecture pattern has been much in discussion, it has not seen commensurate adoption in industry (Paolo et al. 2015). We propose use of digital twin to drive the adaptation using MRAC pattern of Figure 1 (b). Digital twin serves as the Model *i.e.*, reference behaviour of the system. The sensory framework and the pattern-matching infrastructure of ESL constitute the Monitoring and Sense Making machinery. The MAPE-K like infrastructure discussed above can provide the Controller.

4 REAL WORLD USE CASES

We have been applying the proposed approach to a variety of enterprise scenarios spread across business domains (Barat et al. 2017b, Allison et al. 2006, Barat et al. 2019). We discuss below two of these cases that have reached industry adoption stage. They illustrate the essence of the proposed approach while not covering it in its entirety.

4.1 Stock Replenishment in a Retail Chain

4.1.1 Problem

Consider a retail supply chain selling 'P' product types through 'S' stores whose stock is replenished every 'h' hours in a day. The goal is to compute inventory replenishment quantity, for each product in each store, for the retailer. This computation is to be done while optimizing the supply chain as a whole *i.e.* stores, logistics, transportation, warehouses etc and not solely focusing on the extremities *i.e.* the stores. The number of decisions to be computed is huge: with typically 'P' in tens of thousands and 'S' in thousands result in tens of million decisions every 'h' hours. Since there are many capacity sharing points such as warehouse labour, truck volume *etc.*, all these decisions are interdependent. As a result, humans cannot take such decisions quantitatively, and have to rely on gut feel. The state of the art is to either (i) take the

decisions manually, assuming all products are independent, or (ii) encode the business rules in the form of heuristics. In both cases, the stores are kept well stocked, but every other node in the supply chain takes a hit. Moreover, both approaches rely on past data (*i.e.*, consider only what has happened and chose to ignore what could have happened) and aggregated heuristics that connote average case behaviour (*i.e.* not the outliers). As a result, both lead to sub-optimal performance. Also, stock replenishment problem has three dimensions namely shop, warehouse and logistics connecting the two thus requiring an integrated solution covering all three dimensions. Current practice relies on analysis in local contexts, for instance, shop, warehouse and logistics which can lead to optimal solutions for the three contexts, however, it provides little or no help in combining the three into a holistic solution and also identifying a-priori the ramifications of decisions taken to achieve local optimality on global robustness.

4.1.2 Solution

Use of digital twin helps overcome these lacunae. Actor-based nature of digital twin leads to a fine-grained model where every influencing entity is represented as an actor thus leading to an 'in silico' representation that is very close to the reality in terms of the micro-behaviours. The digital twin can be subjected to a variety of situations through what-if scenario playing thus generating more comprehensive data which facilitates better decisions - in local as well as global contexts. Moreover, the decisions can be validated using digital twin before implementing in the real system. Also with accurate capture of actor interactions leading to high-fidelity macro-behaviour, it can be analyzed for undesirable patterns thus ensuring global robustness with local optimality. Moreover, every decision (or decision recommendation) can be justified in quantitative terms.

We have addressed this need for a subset of a retail supply network consisting of a set of distribution centers supplying goods to shops as shown in Figure 6 (a). The objectives of the system are: (i) to ensure stock availability, (ii) to optimize store space, (iii) to minimize wastage of perishable goods, and (iv) to minimize transportation costs. The solution needs to take into account the dynamics of shop-specific consumption behaviour, perishable nature of goods, carrying capacity of trucks, truck routes etc.

We used an actor model based simulation framework for sufficiently training a RL agent and validating new policy in a synthetic environment as shown in Figure 6 (b). The proposed framework contains two control loops: (i) a model centric loop for training of Reinforcement Learning (RL) agent and evaluation of new policies prior to their implementation in a real system, and (ii) real time control loop that include real system. We consider an extended form of actor model to closely mimic the complex systems; adopt simulation as an aid to compute micro-behaviours and observe emerging macro behaviours, overall system state and rewards; and use RL agent (or controller) as the primary computing machinery for deriving suitable actions over time.

We used ESL to create a digital twin of the supply chain network where all subsystems and elements are represented using Actors, where each actor has its own state, trace, and autonomous, self-organizing, and probabilistic behaviours. They may change their states based on time (*e.g.*, product expiry), external events (*e.g.*, on receipt of Order(Products)), and other spatio-temporal conditions (*e.g.*, product packaging and transportation delays). The overall state emerges from multiple micro-behaviours and their interactions that include product order, damages, product expiry and order arrivals. We trained an RL agent to compute actions to maximise long-term rewards in emerging scenarios using the digital twin. As an evaluation, we specify and initialise the characteristics of the simulation and reinforcement learning framework, using a data set spanning one year derived from a public source. A total of 220 products were chosen from the data set, and their meta-data (volume, weight, shelf-life which was not available in the original version) was input manually. A single store and a single truck was used for this case study, with the time between successive delivery moments set to 6 hours (leading to 4 DM per day) with lead time as 3 hours.

Digital twin was subjected to a set of relevant what-if scenarios thus generating more comprehensive training input for RL-based controller. It resulted in reduced training time as well as improved controller performance.

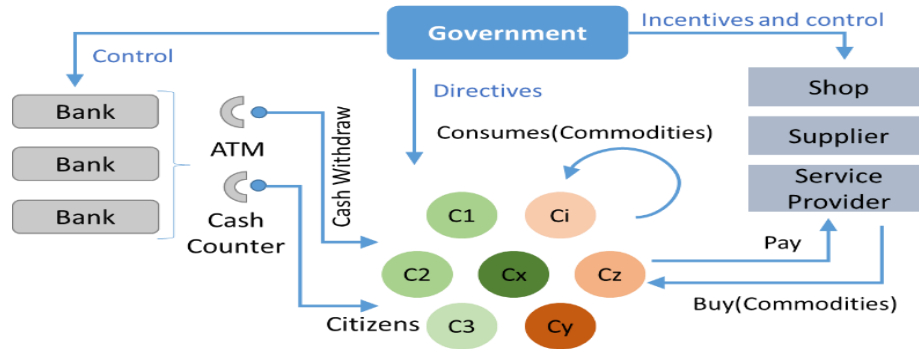


Figure 8. Indian demonetization - subset of society modeled as a digital twin.

Figure 7 (a) shows that the reward at the end of RL-agent training exercise exceeds the heuristic performance, and this advantage is retained on the test data set as well (plotted using separate markers at the ends of the curves). Figure 2 (b) also depicts the ‘exploration rate’ of RL, which is the probability with which the RL algorithms takes randomised actions (to explore the feature space). This rate is brought to zero towards the end of training, and is also zero for testing.

Figure 7 (b) depicts the performance advantage of RL over the heuristic as regards inventory. Though both algorithms begin with an initial (normalised) inventory level of 0.5 for all products, RL is able to maintain a higher average inventory level than the heuristic, by taking advantage of features such as the unit volumes and weights of individual products.

4.2 Demonetization in Indian Economy

4.2.1 Problem

The study is about the recent Demonetisation initiative in India. The cash in circulation in Indian economy had increased significantly over the years (Government of India 2016) and the cash in circulation was 15.4 trillion rupee notes in November 2016. This led to an undesirable shadow economy and funds used for illegal activities. As a corrective action, the Indian government announced the demonetisation of large denomination notes on 8th November 2016 wherein the 87% cash in circulation was removed from the economy with a plan to replenish the cash in a controlled manner (Wikipedia 2016). Limitations were imposed on the exchange of old notes, ATM withdrawal, and daily bank withdrawals to control the negative impacts of the demonetisation.

The sudden nature of the demonetisation event, the incomplete knowledge about possible consequences, and unforeseen behaviour of citizens made an impact on the economy in the weeks that followed. Citizens were inconvenienced and often economically threatened due to the prolonged cash shortages. The government tried to minimise the impacts of the demonetisation by monitoring the situation in real-time and adopting new courses of action on the fly.

This case study was about recreating the demonetization situation ‘in silico’, play out various interventions introduced by the government as what-if scenarios on the digital twin, and get a feel for the efficacy of these interventions in restoring normalcy.

4.2.2 Solution

We created a digital twin for this socio-techno-econo system. We considered a small but well-formed subset of society affected by demonetisation thus limiting the focus to common Indian citizens, who are largely confined to a bounded set of activities, as shown in the Figure 8. Essentially, citizens consume essential

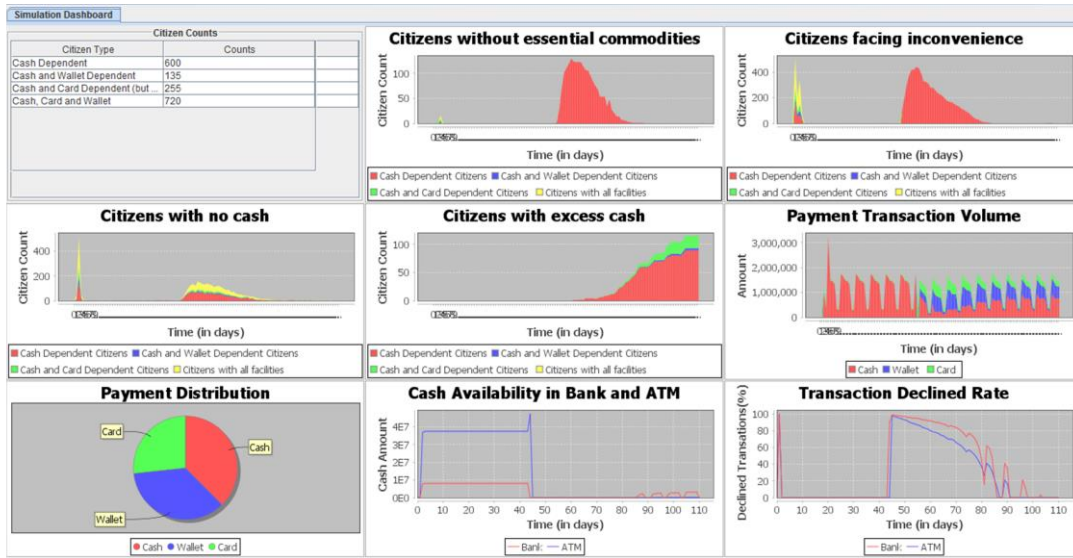


Figure 9. Simulation Dashboard.

and/or luxury commodities (*e.g.*, food, medicines, perfumes *etc.*), and avail various services (*e.g.*, medical assistance, hospitality services, fitness related services *etc.*). They buy commodities from shops/suppliers, avail services from service providers, and pay for their purchases and services. Citizens withdraw cash when cash-in-hand dips below a threshold value. A class of citizens may hold credit and/or debit cards - a citizen who holds card may choose to pay by cash or by card for a purchase, and may withdraw cash from ATM machine and/or bank counter. In contrast, a citizen without a card always pays by cash and withdraws cash from bank counters. We assume all citizens are able to satisfy their daily needs *i.e.*, we excluded poverty and other societal conditions from our experiments.

Pre-demonetization stage is characterized by sufficient cash in ATMs and Banks to service their customers (*i.e.*, citizens), sufficient stock in shops, and no notable denial of service from banks and ATM machines (*i.e.*, citizens are able to withdraw cash when in need). We consider this condition as normal situation. Demonetisation event disrupts this normalcy with abrupt elimination of 86% cash from the economy with a plan to slowly restore cash levels back to 70% of pre-demonetisation level.

To manage the disruption, Banks enforced restrictions on cash withdrawals (amount as well as frequency of withdrawal) right after the demonetisation event principally to ensure fair distribution of new currency notes being introduced at a fixed rate – a mint-centric constraint. Shops adapted to the reduced cash in the economy by beginning to accept alternate payment options such as mobile wallet and card payment whenever they observed a drop in sales record. Citizens adopted to the disruption by changing behavior along two dimensions:

Payment Pattern: Citizens started using mobile wallet and/or card as a payment option to save the trouble of standing in long queues to withdraw cash. However, not everyone used alternate option, an individual's decision were based on several factors such as availability and familiarity with payment technology, and whether the citizen was an early or late adopter to the new technology.

Cash Withdrawal Pattern: Some citizens restored to temporary hoarding of cash *i.e.*, withdrawing cash way in excess of their needs. There could be several rational and irrational reasons behind this panic behaviour.

We simulated a society with one government, one bank, 15 shops and 1710 citizen actors for 150 days, where the first 15 days are considered for setup phase, next 30 days are the pre-demonetisation phase, and 105 days are the post-demonetisation phase. A snapshot of simulation dashboard with operational graphics at the day of 115 days (*i.e.*, after 70 days of demonetisation) is depicted in Figure 9.

We observe that the graphs are unstable for first 15 days of simulation run as simulator is trying to set the values based on actor behaviours and their interactions. The simulation outcome for pre-demonetisation phase is stable and normal: no bank withdrawal request is denied, no citizen is facing any financial crisis, and citizens are not experiencing any deficiency for essential or luxury items. The demonetisation event is triggered at day 45 causing a sudden reduction of 86% cash from the bank and ATM machines. Subsequently, the withdrawals from bank and ATM decline whilst wallet payment and card payment increase significantly: the citizens have started facing a financial crisis and the citizens who are solely dependent on cash have started starving for essential and/or luxury items. The adverse effects continue for almost 50 days and then the situation returns to normal.

In graph with title ‘Citizen with excess cash’ in Figure 9, we observe 115 citizens are hoarding cash when the situation is on the verge of returning back to normal. We also observe that cash dependent citizens are more prone to cash hoarding behaviour. The ‘Payment Transaction Volume’ chart describing the history of overall payment transactions shows an interesting trend – the card (green) and wallet (blue) usage have increased in first 30-40 days of post-demonetisation phase, and then it slowly started reducing.

We correlated these simulation observations with the information found in authentic press-releases and newspapers. The trends on cash conditions of different citizens (shown in ‘Citizen with no Cash’ and ‘Citizen with excess Cash’ charts in Figure 9), the inconvenience due to deficiency of essential items (shown in chart ‘Citizens without essential commodities’ in Figure 9) and luxury items (shown in chart ‘Citizen facing inconvenience’ in Figure 9) for cash dependent citizens, and service of denial at Bank and ATM withdrawal are in tune with the reality. In reality, the cash conditions in ATMs and Banks at the end of January 2017 (after 3 and half months of demonetisation) were just sufficient to serve their customers - this observation related with the graph shown in ‘Cash Availability in Bank and ATM’ graph of Figure 9. The model construction, validation and detailed simulation results can be found in (Barat et al. 2017b).

5 SUMMARY

This paper argued that analysis, control and adaptation have emerged as the principal concerns faced by modern enterprises. It discussed how enterprise modeling state of the art is inadequate for addressing these concerns. It presented an approach that integrates Actor model, reinforcement learning, bottom up simulation and model reference adaptive control to support adaptive enterprises using hi-fidelity digital twin. Two real life case studies highlighting efficacy of the approach and supporting technology were described.

REFERENCES

- Allen, J. (2013). *Effective Akka*. O'Reilly Media, Inc.
- Allison, J., Backman, D. and Christodoulou, L. (2006). “Integrated Computational Materials Engineering: A New Paradigm for the Global Materials Profession”. *Journal of the Minerals, Metals and Materials Society*. 2006 Nov 1;58(11):25-7.
- Armstrong, J. (1996). “Erlang - a Survey of the Language and its Industrial Applications”. In *Proceedings of the symposium on industrial applications of Prolog (INAP)*, page 8.
- Bar-Yam, Y. (1997). *Dynamics of Complex Systems (Vol. 213)*. Reading, MA: Addison-Wesley.
- Barat, S., Kulkarni, V., Clark, A. and Barn B. (2017). “An Actor-model Based Bottom-up Simulation - An Experiment on Indian Demonetisation Initiative”. In *Proceedings of the 2017 Winter Simulation Conference* edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page. IEEE Press: 860-871.
- Barat, S., Kadhilkar, H., Meisheri, H., Kumar, P., Gajrani, M., Baniwal, V. and Kulkarni, V. (2019). “Actor Based Simulation for Closed Loop Control of Supply Chain using Reinforcement Learning”. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS 2019: 1802-1804.
- Butler, H., Hondred, G., van Amerongen, J. (1992). *Model Reference Adaptive Control: Bridging the Gap from Theory to Practice*. <http://resolver.tudelft.nl/uuid:2d0e936c-78f6-4bb0-8fb7-fd07db9cdcbcb> (last accessed - August, 2019)
- Candes, E. J. and Tao, T. (2005). “Decoding by Linear Programming”. *IEEE transactions on information theory*, 51(12):4203–4215.
- Clark, A., Kulkarni, V., Barat, S., Barn, B. (2017). “ESL: An Actor-Based Platform for Developing Emergent Behaviour Organisation Simulations”. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, Cham, 2017 PAAMS 2017: 311-315

Kulkarni, Clark, and Barat

- Clark, A., Barn, B., Kulkarni, V. and Barat, S. (2017). "Querying Histories of Organisation Simulations". Association for Information Systems (AIS) Library.
- Currall, S. C. and Towler, A. J. (2003). *Research Methods in Management and Organizational Research: Toward Integration of Qualitative and Quantitative techniques*. Sage Publications.
- Gosavi A. (2003). "Simulation-based Optimization. Parametric Optimization Techniques and Reinforcement Learning". *Interfaces*, 35(6), 535
- Government of India. (2016). Notes in Circulation, India. <https://data.gov.in/resources/statistics-notes-circulation-india-2001-2015/download> (last accessed - August, 2019).
- Haller, P. and Odersky, M. (2009). "Scala actors: Unifying Thread-based and event-based programming". *Theoretical Computer Science*, 410(2):202–220.
- Hewitt, C., Bishop, P. and Steiger, R. (1973).. A Universal Modular ACTOR formalism for artificial intelligence. In *Proceedings of the 3rd International Joint Conference on Artificial intelligence* (pp. 235-245). Morgan Kaufmann Publishers Inc..
- Iacob, M., Jonkers, D. H., Lankhorst, M., Proper, E., and Quartel, D. D. (2012). *ArchiMate 2.0 Specification*. The Open Group. Van Haren Publishing.
- John, H., Rouncefield, M., and Whittle, J. (2011). "Model-driven Engineering Practices in Industry". In *Proceedings of the 33rd International Conference on Software Engineering*, pp. 633-642. ACM.
- Jonkers, H., Lankhorst, M., Van Buuren, R., Hoppenbrouwers, S., Bonsangue, M. and Van Der Torre, L. (2004). "Concepts for Modeling Enterprise Architectures". *International Journal of Cooperative Information Systems*. 13(03), pp.257-287.
- Khadilkar H. (2018). "A Scalable Reinforcement Learning Algorithm for Scheduling Railway Lines". *IEEE Transactions on ITS*.
- Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International.
- Krogstie, J. (2008). "Using EEML for Combined Goal and Process Oriented Modeling: A Case Study". In *Proceedings of EMMSAD'08*. Thirteenth International Workshop on Exploring Modeling Methods for Systems Analysis and Design.
- Kulkarni V, Clark A and Barn B. (2014). "A Component Abstraction for Localized, Composable, Machine Manipulable Enterprise Specification". *BMSD Conference*, Luxembourg.
- Kulkarni, V., Barat, S., Clark T. and Barn B. (2017). "Supporting Organisational Decision Making in Presence of Uncertainty". *The European Symposium on Modeling and Simulation (EMSS 2017)*.
- Kulkarni, V. (2016). "Model Driven Development of Business Applications: a Practitioner's Perspective". *ICSE (Companion Volume) 2016*: 260-269.
- Kulkarni, V. (2010). "Raising Family is a Good Practice". *FOSD 2010*: 72-79.
- Kulkarni, V. and Barat S. 2011. "Business Process Families using Model-driven Techniques". *IJBPIIM* 5(3): 204-217 (2011).
- Kulkarni, V. and Reddy, S. (2008). "An Abstraction for Reusable MDD Components: Model-based Generation of Model-based Code Generators". *GPCE 2008*: 181-184
- Mcmillan, C. J. (1980). "Qualitative Models of Organisational Decision-making". *Journal of General Management*, 5(4):22–39.
- Meadows, D. and Wright, D. (2008). "Thinking in Systems: A primer". *Chelsea Green Publishing*.
- Michalski, R. S. (1993). "Inferential Theory of Learning as a Conceptual Basis for Multistrategy Learning". *Machine Learning* 11.2-3: 111-151
- Paolo, A., Riccobene, E. and Scandurra, P. (2015). "Modeling and analyzing MAPE-K feedback loops for self-adaptation." In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-managing Systems*, pp. 13-23. IEEE Press, 2015.
- Schrijver, A. (1998). *Theory of Linear and Integer Programming*. John Wiley & Sons.
- TCS (2019). TCS MasterCraft. <https://mastercraft.tcs.com/> (last accessed August – 2019)
- Thomas, M. and McGarry, F. (1994). "Top-down vs. Bottom-up Process Improvement. *IEEE Software*, 11(4):12–13.
- White, S. (2004). "Introduction to BPMN". *IBM Cooperation*, 2(0).
- Wikipedia, 2016. 2016 Indian banknote demonetisation. https://en.wikipedia.org/wiki/2016_Indian_banknote_demonetisation (last accessed - August, 2019)
- Yu, E., Strohmaier, M. and Deng, X. (2006). "Exploring Intentional Modeling and Analysis for Enterprise Architecture". In *Enterprise Distributed Object Computing Conference Workshops*. EDOCW'06. 10th IEEE International (pp. 32-32).
- Zachman, J.A. (1987). "A Framework for Information Systems Architecture". *IBM systems journal*, 26(3), pp.276-292.

AUTHOR BIOGRAPHIES

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. His email address is vinay.vkulkarni@tcs.com.

TONY CLARK is Professor at Aston University in the UK. His academic research interests include programming languages, enterprise modeling, simulation and reinforcement learning. His email address is tony.clark@aston.ac.uk.

SOUVIK BARAT is Senior Scientist at Tata Consultancy Services Research. His research interests include simulation, agent and actor technology, enterprise modeling and digital twin. His email address is souvik.barat@tcs.com.