Towards Adaptive Enterprises using Digital Twins

Vinay Kulkarni
Tata Consultancy Services Research, India

Tony Clark
Aston University, UK

Abstract— *Modern enterprises are large complex systems operating in highly dynamic environments thus requiring to respond quickly 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 outline a knowledge-guided simulation-aided data-driven model-based evidence-backed approach 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. We describe the core modeling and model processing infrastructure developed, and 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.*

Keywords— *Enterprise Modeling, Simulation, Decision Making, Actors, Enterprise Information Systems, Design & Control, Adaptation*

Proposed length of the tutorial *– 3 hours*

Level of the tutorial *– Introductory*

Target audience *– Academic as well as industrial researchers, students, and industry practitioners having an exposure to information systems and modeling in general.*

The presenters

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 work in model-driven software engineering has led to a toolset that has been used to deliver several large business-critical systems over the past 20 years. Much of this work has found a way into OMG standards. Vinay also serves as Visiting Professor at Middlesex University London.

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.

I. Motivation

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[1]. 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 automated 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 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-, effort- and intellectually-intensive endeavour.

To address the problem of enterprise adaptation, in this tutorial, we propose 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.

II. Overview of the Tutorial

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.

A. Line of attack

[1] http://www.valueteam.biz/why-72-percent-of-all-business-transformation-projects-fail

978-1-7281-4844-1/19/$31.00 ©2019 IEEE

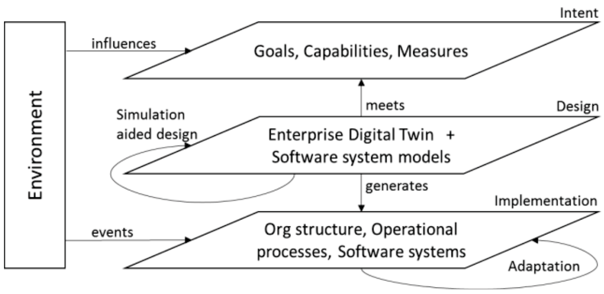

Fig 1. Adaptive Enterprise

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 Fig 1. 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 (MRAC) [1] shown in Fig 2. 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

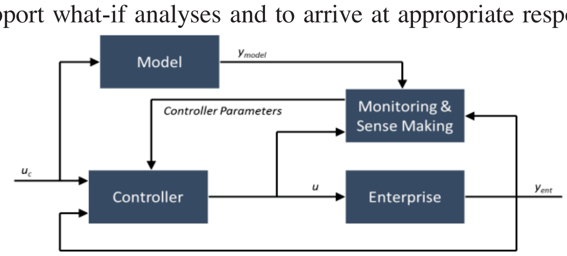

Fig 2. Model driven adaptive enterprise

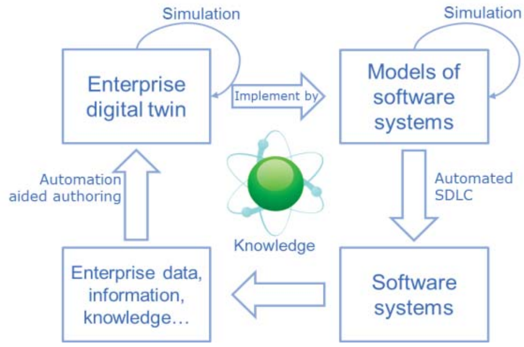

Fig 3. Model-based architecture for adaptive enterprise

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 [2]. 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 [3]. 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 3 provides a pictorial representation of proposed line of attack that hinges on: (i) model-based machinery to help define enterprise digital twin, (ii) simulation machinery what-if and if-what analysis, (iii) model-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.

B. Enterprise Digital Twin

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. Majority of these languages can be traced to Zachman framework [4] 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 automated analysis as the information is captured typically in the form of texts and pictures. It can be observed of the existing EM languages that: the languages capable of specifying all the relevant aspects of

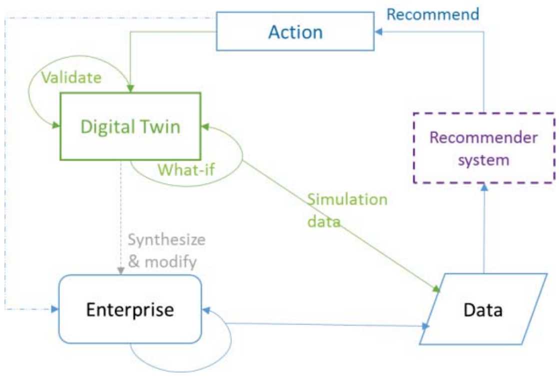

Fig 4. Assured recommendation using digital twin

an enterprise for organisational decision-making lack support for automated analysis [4, 5, 6], and the languages capable of automated analysis can only address a subset of the aspects required for decision-making [7, 8, 9]. 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 a mathematical 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) [10]. We have developed an Actor-based modelling language, ESL[2], to specify an enterprise as a set of autonomous encapsulating units that interact with each other by exchanging messages [11]. We have developed a component abstraction to model the fractal nature of system of systems [12]. We have extended the actor-based paradigm to enable modelling of uncertainty [13]. Through a set of case studies, we have validated the adequacy of ESL to specify complex system of systems for decision-making [14, 15].

C. Using the Digital Twin for Design and Control

An Adaptive Enterprise design process aims to use information to nudge observed measures to achieve designed goals. Domain knowledge and experience are captured in a set of purposive models i.e. Digital Twin. The digital twin facilitates the design process through simulation. It is essentially an iterate-till-saturate process wherein one starts with an initial configuration of capabilities and progressively nudges the measures using levers till the desired objectives are met. We use pattern matching to extract measures of interest from the execution trace produced by simulation to check whether the goals are met. We use machine learning techniques to analyze the effects of modifications introduced to measures as regards achieving the desired goal. This learning element leads to continuous improvement of the design process over time.

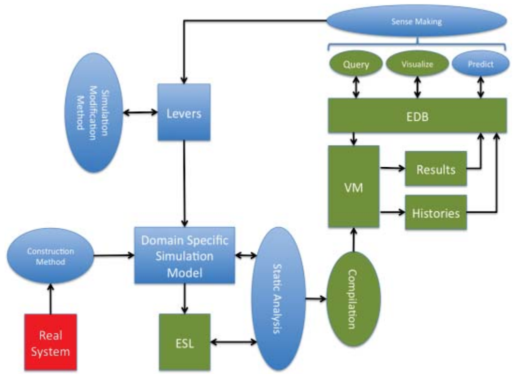

Fig 5. ESL Workbench

Decision-making processes involve analyzing past data to come up with a set of recommendations for achieving the desired objectives. In a complex system of systems, one only has visibility of a local context in arriving at such recommendations. One would like to know a-priori the ramifications of a recommendation at the system level – whether it will achieve the desired goal, whether it will adversely impact other goals *etc*. A digital twin helps answer these questions by simulating the recommendation as shown in Fig 4. In addition, a digital twin can help improve the decision-making process itself by exploring the solution space through what-if scenario playing. Moreover, over time, synthesis of these explorations can help improve the overall operation of the system itself.

D. The ESL Workbench

The ESL workbench[3] supports an Actor-based language for simulation-based decision-making [11]. The workbench is shown in Fig 5 and will be introduced as part of the tutorial. In order to provide a sense of the workbench features we describe a simple actor-based simulation that is represented, run and analysed using the key technologies. Fig 6 shows a shop with customers, assistants and tills. The shop owner has a goal of minimising disgruntled customers who leave the shop because they wait too long in queues to be served at tills or provided with help. Customers and assistants behave autonomously and there are different types of drivers for customer behaviour.

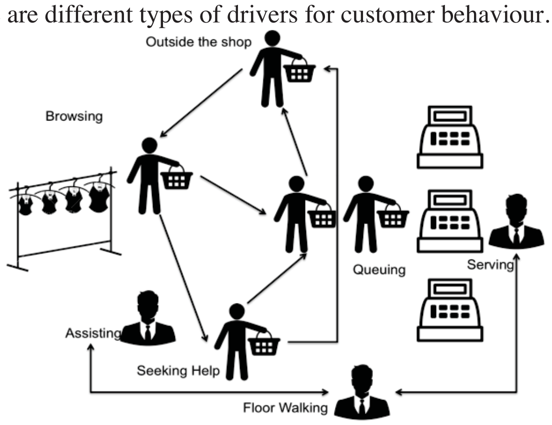

Fig 6. Shop Simulation

[2] http://www.esl-lang.org

[3] http://tonyclark.github.io/ESL/

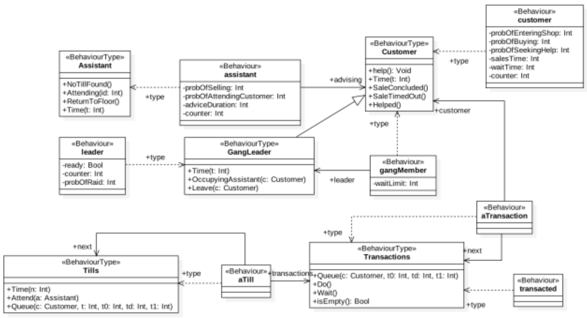

Fig 7. ESL Actor definitions

Figure 7 shows a model of actors that can be directly implemented in ESL. 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 [16]. 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.

E. 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 [17, 18]. 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 [19, 20]. Model based code generators also benefit from model based code generation thus further improving productivity and agility [21]. Several manifestations of this technology exist[4], 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-n-respond architectures.

Though MAPE-K architecture pattern has been much in discussion, it has not seen commensurate adoption in industry

[4] https://mastercraft.tcs.com/

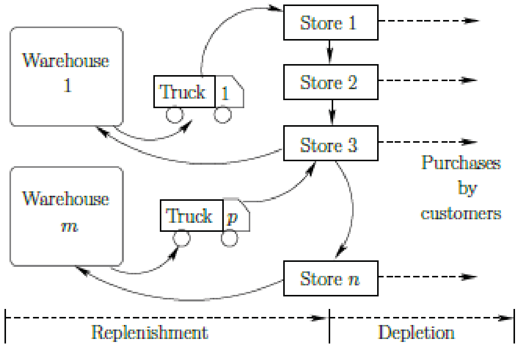

Fig 9. Schematic of stock replenishment in a supply chain

[22]. We propose use of digital twin to drive the adaptation using MRAC pattern of Fig 2. 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 & Sense Making* machinery. The MAPE-K like infrastructure discussed above can provide the *Controller*.

F. Real World Use-cases

The tutorial includes several real-world use-cases demonstrating how adaptiveness can be imparted using digital twin models and associated processing machinery. Description of one such use-case follows.

Retail supply network is a complex system of systems comprising of manufacturing plants, wholesalers, distribution centers, retailers, stores, logistic providers etc. These actors are typically designed to operate in a manner so as to achieve their individual goals in a dynamic environment. Several solutions exist to help design such actors, however, there is no support available to tell a-priori the ramifications of local changes throughout the system. In other words, there is a need for a controller to guide a complex system of systems operate in a manner that is optimal in local contexts and yet robust overall. We have addressed this need for a subset of the retail supply network consisting of a set of Distribution Centers supplying goods to Shops as shown in Fig 8. 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.

Existing literature that describes the use of Reinforcement Learning (RL) for controlling complex systems [23, 24], typically does so for systems which can be modelled analytically, simplifying the computation of step rewards and next state of the system. This is because RL is effective only when the (stochastic) transition functions closely approximate the real system to be controlled. In situations where the system cannot be described analytically, algebraic expressions cannot be used to compute rewards and transitions. An experimental approach can be used for training the RL agent when the system is non-physical (for example, is itself a piece of software as in the case of computer games) [25]. However, training on the actual system is not feasible in the case of business-critical systems. Therefore, the development of (and integration with) a high-fidelity simulation model is crucial for effective training of the RL agent and controlling complex systems.

We created a digital twin of the supply network and simulated scenarios of interest to generate data to drive reinforcement learning for replenishment decisions at each shop. Thus, the digital twin helped realize a more comprehensive controller by enabling faster tuning of the same. This solution is being validated in a real life supply network [26].

III. Novelty of the Tutorial

This tutorial will introduce a new use of modeling to the conference. Tutorials at agent-based conferences often address simulation. The application of this field to enterprises and placing it in the context of information systems is novel.

This tutorial will introduce delegates both to a novel conceptual approach and to a novel technology platform that are designed to support adaptation through simulation-aided design and decision-making. In addition the delegates will be introduced to case studies that have been drawn from the real world and will therefore be of interest to both academics and industry.

IV. History of the Tutorial

Research underlying this tutorial has been evolving over past 3-4 years. We have presented it in tutorial form at several international conferences such as ESM 2017, ISEC 2018, MoDELS 2017 and MoDELS 2018 where it has been received well.

V. Required Infrastructure

The tutorial will require a standard room with a beamer. The ESL Workbench and example simulations will be made available to delegates.

VI. Sample Slides

Please refer for sample slides here.

References

[1] Butler, H., Hondred,G., van Amerongen, J., 1992, Model Reference Adaptive Control: Bridging the gap from theory to practice. 1992.

[2] Bar-Yam, Y., 1997. Dynamics of complex systems (Vol. 213). Reading, MA: Addison-Wesley.

[3] Gosavi A. Simulation-based optimization. parametric optimization techniques and reinforcement learning. 2003.

[4] Zachman, J.A., 1987. A framework for information systems architecture. IBM systems journal, 26(3), pp.276-292.

[5] Krogstie, J., 2008. Using EEML for Combined Goal and Process Oriented Modeling: A Case Study,[w:] T. Halpin, J. Krogstie, E. Proper. In Proceedings of EMMSAD'08. Thirteenth International Workshop on Exploring Modeling Methods for Systems Analysis and Design.

[6] 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.

[7] Yu, E., Strohmaier, M. and Deng, X., 2006, October. Exploring intentional modeling and analysis for enterprise architecture. In Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International (pp. 32-32). IEEE.

[8] Meadows, D. and Wright, D., 2008. Thinking in systems: A primer. Chelsea green publishing.

[9] White, S, 2004. Introduction to BPMN. IBM Cooperation, 2(0).

[10] Hewitt, C., Bishop, P. and Steiger, R., 1973, August. 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..

[11] Tony Clark, Vinay Kulkarni, Souvik Barat, Balbir Barn. ESL: An Actor-Based Platform for Developing Emergent Behaviour Organisation Simulations. PAAMS 2017: 311-315

[12] Vinay Kulkarni, Tony Clark and Balbir Barn, 2014. A Component Abstraction for Localized, Composable, Machine Manipulable Enterprise Specification. BMSD Conference, 24-26 Jun 2014, Luxembourg.

[13] Vinay Kulkarni, Souvik Barat, Tony Clark and Balbir Barn, 2017. Supporting Organisational Decision Making in Presence of Uncertainty. The European Symposium on Modeling and Simulation (EMSS 2017).

[14] Souvik Barat, Vinay Kulkarni, Tony Clark, Balbir Barn, 2017. An actor-model based bottom-up simulation - An experiment on Indian demonetisation initiative. WSC 2017: 860-871.

[15] Souvik Barat, Asha Rajbhoj, Prashant Kumar, Vinay Kulkarni, 2017. A Case Study Exploring Suitability of Bottom Up Modelling and Actor-based Simulation for Decision Making. DIAS/EDUDM@ISEC 2017.

[16] Tony Clark, Balbir Barn, Vinay Kulkarni, Souvik Barat, 2017. Querying Histories of Organisation Simulations. ISD 2017.

[17] Vinay Kulkarni, 2016. Model driven development of business applications: a practitioner's perspective. ICSE (Companion Volume) 2016: 260-269.

[18] Hutchinson, John, Mark Rouncefield, and Jon Whittle. "Model-driven engineering practices in industry." In Proceedings of the 33rd International Conference on Software Engineering, pp. 633-642. ACM, 2011.

[19] Vinay Kulkarni, 2010. Raising family is a good practice. FOSD 2010: 72-79.

[20] Vinay Kulkarni, Souvik Barat, 2011. Business process families using model-driven techniques. IJBPIM 5(3): 204-217 (2011).

[21] Vinay Kulkarni, Sreedhar Reddy, 2008. An abstraction for reusable MDD components: model-based generation of model-based code generators. GPCE 2008: 181-184

[22] Arcaini, Paolo, Elvinia Riccobene, and Patrizia Scandurra. "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.

[23] T Gabel and M Riedmiller. 2012. Distributed policy search reinforcement learning for job-shop scheduling tasks. International Journal of Production Research 50, 1(2012), 41–61.

[24] H Khadilkar. 2018. A Scalable Reinforcement Learning Algorithm for Scheduling Railway Lines. IEEE Transactions on ITS (2018).

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013).

[26] H Khadilkar, Souvik Barat, Vinay Kulkarni…2019. Actor Based Simulation for Closed Loop Control of Supply Chain using Reinforcement Learning. International Conference on Autonomous Agents and Multiagent Systems, 13-17 May 2019, Montreal, Canada (2019).