ESL: An Actor-Based Platform for Developing Emergent Behaviour Organisation Simulations

Tony Clark[1], Vinay Kulkarni[2], Souvik Barat[2], and Balbir Barn[3]

[1] Sheffield Hallam University, UK
[2] Tata Consultancy Services Research, India
[3] Middlesex University, UK

1 Abstract

Specification and analysis of complex systems can be approached top-down or bottom-up [11]. A top-down approach conceptualises a system using a global state and the behaviour represented using an aggregated macro-behaviour of the system elements. For example, the System Dynamics (SD) model [8] uses the concepts of stocks, flows, feedback loops and time delays. A top-down approach considers a reductionist view [10] to understand system using the mathematical rigour from operational research, optimization theory, and sophisticated AI algorithms. A bottom-up approach, in contrast, considers the micro-behaviour of individual elements and their interactions. Conceptually, the bottom-up approach relies on emergentism [9] as advocated in actor model of computation [1], and agent-based systems [6].

Top-down approaches are a popular choice for analysing and understanding complex systems in the context of critical business needs such as decision making activities. Existing modelling and analysis tools that support top-down approaches are extremely efficient for describing and simulating the aggregated system behaviour. However, we propose that they are not appropriate for precise understanding of complex and dynamic systems that can only be understood in terms of emergent behaviour, for example systems that contain large numbers of socio-technical [7] elements having adaptive, autonomous and dynamic behaviours.

Our recent work in this area [4, 2, 5, 3] has performed a domain analysis of organisations with socio-technical characteristics with a view to simulation and analysis leading to improved decision-making. This work has led us to choose a bottom-up approach for simulating emergent behaviour based on concepts represented using an actor-based model of computation. Our project has developed a conceptual model for simulation based on goals, measures, levers and adaptation and we are working on an actor-based simulation platform called ESL that supports these concepts. This tool demonstration will show features of ESL in terms of a real-world simulation.

2 Main Purpose

The purpose of this demonstration is to: (1) Introduce the ESL actor-based language and associated development environment. (2) Demonstrate a conceptual approach to the analysis of emergent behaviour. (3) Show how ESL can be applied to a real-world case study.

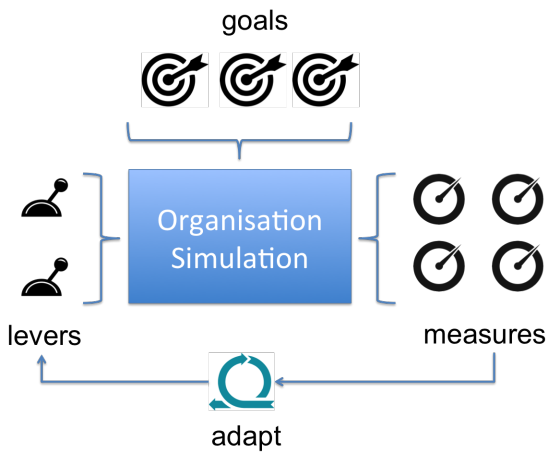

Fig. 1. Conceptual Model for Simulation

3 Demonstration

3.1 ESL

ESL[1] is an text-based language that supports the actor model of computation extended with the following features: pattern-matching over structured data, higher-order functions that can be used to implement complex actor interaction patterns, probabilistic behaviour as required by simulations, and data-locks. ESL execution produces a history that contains a description of actor behaviour. Histories can be visualised using a range of graphical libraries or interrogated using a logic-based query language that is provided by the ESL platform. ESL and ESL simulation engine are written in Java using a virtual-machine.

ESL generates output during execution that and also supports an iterative approach based on histories: (1) identify the actors in the system; (2) model their behaviours; (3) run and capture the history; (4) formulate a theory about the system; (5) list particular theorems that should hold; (6) express each theorem as a query; (7) test that the theorem holds by running the query against the history.

3.2 Conceptual Approach

Figure 1 shows the proposed conceptual model for constructing ESL-based simulations. The goals for decision-making guide the construction of levers that parameterise the simulation which in turn produces measures which are defined by functions over the simulation histories. Adaptation is performed in terms of the histories and the levers.

3.3 Case Study

This demo will use a real-world case study to show the features of ESL for emergent simulation behaviour. The study is taken from the recent Demonetisation initiative in India. The cash in circu-

[1] `https://github.com/TonyClark/ESL`

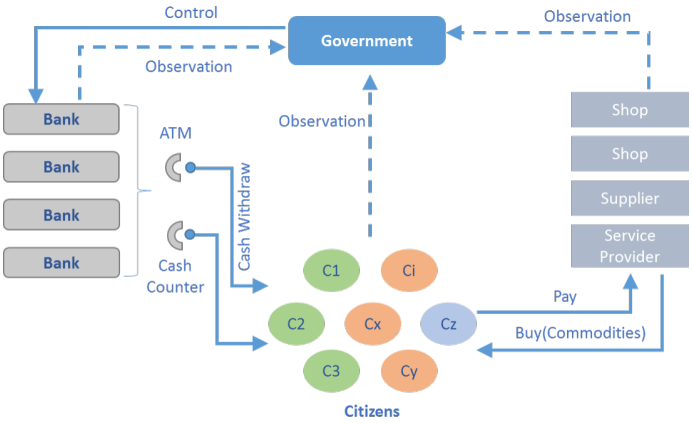

Fig. 2. Overview of Demonetisation Actors

Fig. 3. Simulation Results

lation in Indian economy has increased significantly over the years[2] and the cash in circulation was 17.8 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 November 8 2016 wherein the 86% cash in circulation were pulled out from Indian economy with a plan to replenish the cash in a controlled manner[3]. Limitations were imposed on the exchange of old notes, ATM withdrawal, and daily bank withdrawals to control the negative impacts of the demonetisation.

However, the sudden nature of the demonetisation event, the incomplete knowledge about possible consequences, and unforeseen behaviours of the citizens that emerged due to demonetisation made an impact on the economy in the weeks that followed. The 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.

[2] https://data.gov.in/resources/statistics-notes-circulation-india-2001-2015/download
[3] http://finmin.nic.in/press_room/2016/press_cancellation_high_denomination_notes.pdf

We believe that a simulation based on ESL can help in analysing the efficacy of actions arising from demonetisation. We use an ESL based simulation to understand the impact of demonetisation on a synthetic but near real world Indian society and perform various what-if experiments to explore the implication of various actions. In particular, we use ESL to specify Bank, Citizens, Shops and their interactions as depicted in figure 2 to perform what-if analyses. Figure 3 shows a dashboard snapshot that is produced by ESL from the demonetisation case study.

4 Conclusion and References

ESL and ESL simulation engine are actively being developed. Current plans are to add reasoning to actors in the form of logic programming over local histories, and to develop a monitor language. The former will be used to add intelligence and planning abilities to actors, and the latter will be used to encode adaptation rules.

References

1. Gul Agha. *An overview of actor languages*, volume 21. ACM, 1986.
2. Souvik Barat, Vinay Kulkarni, Tony Clark, and Balbir Barn. A simulation-based aid for organisational decision-making. In Leszek A. Maciaszek, Jorge S. Cardoso, André Ludwig, Marten van Sinderen, and Enrique Cabello, editors, *Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016) - Volume 2: ICSOFT-PT, Lisbon, Portugal, July 24 - 26, 2016.*, pages 109–116. SciTePress, 2016.
3. Balbir S. Barn, Tony Clark, and Vinay Kulkarni. Next generation enterprise modelling - the role of organizational theory and multi-agent systems. In Andreas Holzinger, Thérèse Libourel, Leszek A. Maciaszek, and Stephen J. Mellor, editors, *ICSOFT-EA 2014 - Proceedings of the 9th International Conference on Software Engineering and Applications, Vienna, Austria, 29-31 August, 2014*, pages 482–487. SciTePress, 2014.
4. Tony Clark, Vinay Kulkarni, Souvik Barat, and Balbir Barn. Actor monitors for adaptive behaviour. In Ravi Prakash Gorthi, Santonu Sarkar, Nenad Medvidovic, Vinay Kulkarni, Atul Kumar, Padmaja Joshi, Paola Inverardi, Ashish Sureka, and Richa Sharma, editors, *Proceedings of the 10th Innovations in Software Engineering Conference, ISEC 2017, Jaipur, India, February 5-7, 2017*, pages 85–95. ACM, 2017.
5. Vinay Kulkarni, Souvik Barat, Tony Clark, and Balbir S. Barn. Toward overcoming accidental complexity in organisational decision-making. In *18th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MoDELS 2015, Ottawa, ON, Canada, September 30 - October 2, 2015*, pages 368–377, 2015.
6. Charles M Macal and Michael J North. Tutorial on agent-based modelling and simulation. *Journal of simulation*, 4(3):151–162, 2010.
7. Tom McDermott, William Rouse, Seymour Goodman, and Margaret Loper. Multi-level modeling of complex socio-technical systems. *Procedia Computer Science*, 16:1132–1141, 2013.
8. Donella H Meadows and Diana Wright. *Thinking in systems: A primer*. chelsea green publishing, 2008.
9. Timothy O'Connor and Hong Yu Wong. Emergent properties. 2002.
10. Ralf Stoecker. Emergence or reduction?essays on the prospects of nonreductive physicalism. *Philosophy and Phenomenological Research*, 55(3):701–706, 1995.
11. Martyn Thomas and Frank McGarry. Top-down vs. bottom-up process improvement. *IEEE Software*, 11(4):12–13, 1994.