A Simulation-Based Aid for Organisational Decision-Making

Author names omitted due to the blind review

Keywords: Organisational decision making, Enterprise modelling languages, Meta modelling, Simulation.

Abstract: Effective decision-making of modern organisation often requires deep understanding of various aspects of organisation such as organisational goals, organisational structure, business-as-usual operational processes. The large size of the organisation, its socio-technical characteristics, and fast business dynamics make this endeavor challenging. Industry practice relies on human experts for comprehending various aspects of organisation thus making organisational decision-making a time-, effort- and intellectually-intensive endeavor. We are working on a specific instance of this problem in bespoke software development space. This paper proposes a model-based simulation approach to organisational decision-making. We illustrate how this is applied to a real life problem from software service industry.

1 INTRODUCTION

The modern changing business context requires that organisations make responses such as critical decisions, to a variety of change drivers in order to stay competitive. In order to minimise undesirable consequences such as prohibitive costs of erroneous decisions and lack of opportunities for later course correction, there is a need for a-priori judicious evaluation of the available courses of action (Shapira, 2002). The decision-makers are thus expected to understand, analyze and correlate existing information about various aspects of enterprise such as organisational goals (Shapira, 2002), organisational structure (Parsons and Jones, 1960), business-as-usual operational processes (BAU) (Conrath, 1967; Parsons and Jones, 1960; Locke, 2011), change drivers and their influences on overall organisation. Large and complex organisational structure (Parsons and Jones, 1960), inherent socio-technical characteristics of the organisation (McDermott et al., 2013), dynamic operating environments (Conrath, 1967), and multiple stakeholders with possibly conflicting goals (McDermott et al., 2013) all contribute to the complexity of organisational decision-making.

Current industry practice relies mostly on human experts for decision-making with spreadsheet, word processors, and diagram editors being the most popular tools used for capturing the relevant information about enterprise. The informal nature of the information means the power, rigour, and speed of sophisticated analysis due to automation cannot be brought to bear upon the decision-making problem. As a result, the quality of the solution is largely dependent on knowledge and experience of human experts involved in the decision-making process. When this is coupled with the sheer volume, heterogeneity and location of the information, the complexity of a dynamic environment and the need to keep information up to date then the analysis is further untenable. Provision of solutions that are able to stitch together a coherent, consistent and integrated view from these pieces is challenging for decision-makers.

Enterprise Modelling (EM) try to reduce the complexity of organisational decision-making with a range of concepts, languages and tools for representing and analyzing the aspects of organisation. For instance, the Zachman framework (Zachman et al., 1987) advocates six aspects namely *why*, *what*, *how*, *who*, *when* and *where* for comprehensive representation of an organisation. Thus it can be argued that complete specification of enterprise is possible using Zachman framework, however, no automated analysis support is available. Examination of existing EM reveals some interesting observations. Languages capable of specifying all the relevant aspects of enterprise for organisational decision-making lack support for automated analysis (e.g., Archimate (Iacob et al., 2012), IEM (Bernus and Schmidt, 2006), EEML (Krogstie, 2008), BMM (OMG, 2015), and UEML (Vernadat, 2002)). Languages capable of automated analysis only cater for a subset of the relevant aspects for decision-making (e.g., BPMN (OMG, 2011), i* (Yu et al., 2006), System Dynamics (Meadows and Wright, 2008) and ARIS (Scheer and Nüttgens, 2000)). Co-simulation using a relevant subset of EM languages can be a pragmatic solution (Barjis, 2008). For instance, i* (to specify the *why* aspect)

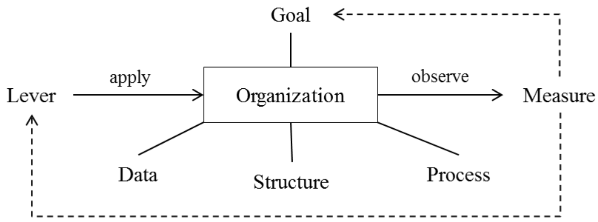


Figure 1: High-level process for organisational decision-making

(Yu et al., 2006), BPMN (to specify the *how* aspect) (OMG, 2011) and Stock-n-flow (to specify the *what* aspect) can be used to come up with the necessary and sufficient specification which is amenable for analysis albeit in parts Human expertise is still required for the analysis of the problem, selection of the appropriate EM technique and the integration of the technology into a consistent whole (Fox, 1994). This intellectual challenge is further exacerbated due to paradigmatically diverse nature of the three languages and issues of interoperability of the various tools.

A simulation-based approach to organisational decision-making may offer a pragmatic solution. However, simulation is known to deliver in situations where mechanistic world view holds (Barjis, 2008) whereas modern enterprises are socio-technical systems (McDermott et al., 2013) bringing additional dimensions such as uncertainty, autonomicity and adaptability to the problem space.

We propose a pragmatic model-based simulation approach for analyzing organisations as socio-technical systems. This analysis centric approach hinges on: (i) the necessary and sufficient information for decision-making to exist in machine processable manner, (ii) machinery for effective processing of this information, and (iii) a method to enable repetitive use of the machinery at the hands of knowledgeable users. This paper addresses tenets (i) and (ii) while hinting at (iii). This paper also describes a model-based realization of the proposed approach. The approach is illustrated using an example from the software services industry. The principal contributions of this paper are: (i) a modelling abstraction for precise and comprehensive representation of organisations as socio-technical systems, (ii) application of a simulation technique to support organisational decision-making through repeatable scenario playing.

We validate our approach using a case study derived from software development activity of a large IT consultancy organisation We illustrate how an organisation can be modeled using the proposed concepts. Decision-making will be supported through simulation of the model in order to evaluate the KPIs/measures such as revenue, profit, and customer satisfaction. We show how simulation can support the a-priori development of strategies to maximize a set of chosen organisational measures.

Table 1: Requirements for organisational decision-making

Classification	Requirement	Description
Aspects	Why	Intentional Specification.
	What	Structural Specification.
	How	Behavioural Specification.
	Who	Responsible human actors.
Size	Composability	An assembly of [sub] organisations.
Socio-Technical Characteristics	Reactive	Reactive to environmental events.
	Intentional	Works towards its objectives.
	Autonomous	Organisation is capable of determining its own course of action.
	Modular	Units are encapsulated.
	Adaptable	Responds to changes by transforming itself.
	Uncertainty	Exhibits probabilistic behavior.
Relationship	Aspect Relationship	Relationship between *why*, *what*, *how* and *who* aspects
Analysis	Machine-processability	Models that are amenable for computational analysis
	Quantitative	Simulation based quantitative analysis
	Qualitative	Simulation based qualitative analysis

2 MOTIVATION

2.1 Organisational decision-making

Organisational decision-making activity is an iterative process for selecting appropriate course of actions for achieving the desired goals of organisation.

The process starts with identifying possible goals or course of actions or both. The successive steps deal with validation of selected course of actions against goals, ranking alternatives options, and selection of alternatives over the other. Our visualization of decision-making is depicted in Fig. 1. As shown in the figure, an *organisation (O)* comprises four basic elements *Data (D)*, *Structure (S)*, *Processes (P)* and *Goals (G)* i.e., O = <D, S, P, G>. The data D describes the current state of organisation and a set of past states of interest, process P describes collection of *Business As Usual* (BAU) behaviors, and goals G specify the desired intention of organisation. Structurally an organisation (i.e., S of O) is a composition of interacting socio-technical units where each unit can further be visualized as <D, S, P, G>tuple. An organisation or its constituent unit manages its goals G; goals G affect organisational BAU behaviours P; and organisational BAU behaviours P is accountable for state change leading to data update D. The available data D determines whether the stated goals G are achieved or not. The meaningful state variables

Table 2: State of the art of Enterprise Modelling and Modelling Framework

EM Models and Frameworks	Why	What	How	Who	Modular	Composable	Reactive	Autonomous	Intentional	Adaptable	Uncertainty	Relationship	Visualization	Processable	Quantitative	Qualitative
Zachman Framework (Zachman et al., 1987)	S	S	S	S	S	N	N	N	S	N	N	N	N	N	N	N
Archimate (Iacob et al., 2012)	S	S	S	S	S	I	S	I	S	N	N	S	S	N	N	N
DoDAF (Wisnosky and Vogel, 2004)	S	S	S	S	S	S	S	I	S	N	N	S	I	N	N	N
MoDAF (MOD, 2005)	S	S	S	S	S	S	I	I	S	N	N	S	I	N	N	N
ARIS (Scheer and Nüttgens, 2000)	I	S	S	S	S	S	S	S	I	N	N	I	S	S_{How}	N	S_{How}
EEML (Krogstie, 2008)	I	S	S	S	N	N	N	N	S	N	N	I	S	N	N	N
i* (Yu et al., 2006)	S	N	N	I	S	S	N	N	S	N	N	I	S	S_{Why}	S_{Why}	S_{Why}
BMM (OMG, 2015)	S	N	I	I	I	N	N	N	S	N	N	I	I	N	N	N
BPMN (OMG, 2011)	N	I	I	S	S_{How}	S_{How}	S	I	N	N	N	N	S	S_{How}	S_{How}	S_{How}
IEM (Bernus and Schmidt, 2006)	N	S	S	S	N	N	N	N	N	N	N	I	I	N	N	N
DEMO (Dietz, 2001)	N	I	S	I	I	I	S	S	I	N	N	I	S	S_{How}	S_{How}	S_{How}
MEMO (Frank, 2002)	S	I	S	S	S	S	S	I	S	N	N	S	S	S_{How}	S_{How}	S_{How}
System Dynamics (Meadows and Wright, 2008)	N	S	I	I	I	N	S	S	N	N	N	N	S	S_{What}	N	S_{What}
Legend: S=Supported, Sx= Supported for aspect X, I =Inadequate, N =No Support																

used for evaluation of goals are called the *Measures (M)*. The *Levers (L)* are appropriate course of actions the decision-maker can take for achieving the stated goals. A lever is essentially specification of a change to structure, process, goals or any combination thereof.

Thus, in this formulation, decision-making is human-guided exploration of design space wherein a set of levers L_{select} from the available levers L are selected for application, their effect on the relevant set of measures is observed, the desired goals are (re)evaluated using the new values of measures - this loop continues till either the desired goals are met or the desired goal is changed thus starting off another iterate-till-saturate process. Critically, the ability to specify influence of a lever on a set of measures is the key. The socio-technical nature of an enterprise and the inability to have complete understanding of the problem space make specification of lever-to-measure influence in pure mathematical terms very hard. Therefore, simulation seems to be the pragmatic recourse. Also, choice of levers and the temporal order of their application determines if the desired goals can be achieved or achieved sub-optimally or not at all. This seems more in the realm of art and intuition.

2.2 Tenets of organisational decision-making

We argue that an organisation can be understood well by knowing *what* an organisation is, *how* it operates and *why* it is so (Barn et al., 2014). Further clarity can be obtained by considering organisational responsibilities and understanding the *who* (i.e., responsible stakeholders) aspect of the organisation. Therefore, we consider the four aspects, namely *what*, *how*, *why* and *who*, as necessary and sufficient for specifying data, structure, process and goals of an organisation. This is broadly aligned with the Zachman framework (Zachman et al., 1987) except for the *where* and *when* aspects which we believe are mostly subsumed within *what* and *how* aspects.

This boils down to two primary requirements for supporting organisational decision-making: (i) the ability to capture *why*, *what*, *how* and *who* aspects of an organisation, in a formal manner and (ii) the ability to perform what-if and if-what analyses of the formal specification. Table 1 provides an overview of the key requirements for organisational decision-making.

2.3 State-of-the-art and industry expectations

Our analysis of the current EM literature revealed that none of the existing EM languages are capable of supporting all the desired characteristics of Table 1. Table 2 summarizes the relative adequacy of the EM languages explored. The relative coverage of EM languages requires that the decision-maker combine subsets of languages and their tools to address their modelling needs. Given that any EM language (and tool) has only a partial view, the decision-maker is forced to employ methods that enable expressing the overall problem into parts that are appropriate for each tool used and then enable subsequent integration of each of these partial views. Such approaches raise considerable problems such as: the inability to set up relationships across partial specifications due to differing underlying meta models and interoperability concerns between tools. Moreover, it is still not possible to represent the socio-technical characteristics of an organisation. Recently there is a trend to revisit the concepts like agent-based modelling and simulation (Bonabeau, 2002; Camus et al., 2015), discrete event modelling and simulation (Wainer, 2009), and system dynamic modelling (Meadows and Wright, 2008) to nurture with socio-technical characteristics of modern organisation. Tools such as AnyLogic[1] and Simudyne[2] combine these concepts with the objective of offering a comprehensive solution. However, they do not support specification of all desired aspects and aspect relationships contained in Table 1. These inadequacies of specification languages and associated processing machinery along with lack of help in selecting the right levers and computing their influence on the relevant set of measures to be used for evaluating goals leads to excessive reliance on human expertise. As a result, the organisational decision-making remains a time, effort and intellectually-intensive endeavour. Decision makers expect help not only in identifying the candidate set of levers to be applied at a given state but also with quantitative as well as qualitative estimation of application of the selected levers towards achievement of the stated goals.

3 PROPOSED SOLUTION

We propose a pragmatic approach to improve the current state of organisational decision-making to help decision-makers to analyze various what-if scenario of a decision-making problem. We now describe a model-based realisation of simulation-based analysis approach. Firstly, we propose a conceptual model for representing the *why*, *what*, *how* and *who* aspects of organisation in a localised relatable manner. Further we refine this conceptual model to an implementation model and provide simulation semantics to enable what-if scenario playing thus enabling human-guided exploration of solution space with enhanced certainty. We argue how the proposed implementation model meets the desired tenets, describe an implementation strategy, and outline a packaging that practitioners may find effective in real-life industry-scale situations.

3.1 Conceptual Model

From an external stakeholder perspective, an organisation can be viewed as something that responds to a set of events as it goes about achieving its stated goals. Organisations consist of many autonomous units, organised into dynamically changing hierarchical groups, operating concurrently, and managing

[1]www.anylogic.com/

[2]www.simudyne.com/

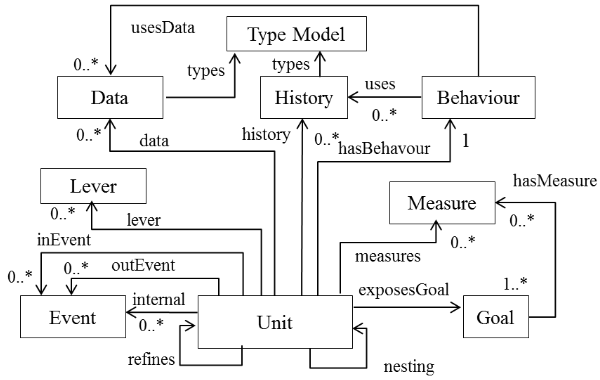


Figure 2: Conceptual model of organisation for decision-making

goals that affect their behaviour. We describe structure and behaviour of an organisation using a small set of concepts and their relationships as depicted in Fig. 2.

3.2 Organisational decision-making

A *Unit* is an autonomous self-contained functional unit with high coherence and low external coupling. It exposes *Goals* stating its intention, and it interacts with environment through a set of *In-Events* and *Out-Events*. Internally it contains a *Behaviour*, a set of *Internal Events* and a *Type Model*. The type model describes the schema for representing current and previous states of the organisation, i.e. *Data* and *History*. A Unit may make use of several *contained* Units in order to meet the promised goals. The contained units can interact with each other to delegate their responsibilities to others; a unit can also participate in hierarchical composition structure to accomplish wider goals of the organisation, e.g., a larger unit or an organisation. A Unit has a set of *Levers* and *Measures* where levers are parameters that can be used for configuration purposes, and measures are meaningful state variables that are exposed to the environment. Conceptually, the elements Unit, unit relation and nesting capability represent the structure S, the Event and Behaviour represent process P, Data and History represent data D, Goal represent the goal G of organisation O. On the other hand elements Unit, Event, Data, History and nesting capability of Unit are capable of specifying the *what* aspect, Goal specifies the *why* aspect, Behaviour specifies the *how* aspect and Unit, as individual, specifies the *who* aspect of an organisation. Event helps to capture reactive nature of Unit, the intent is captured using Goal, modularity is achieved through Unit, autonomy is possible due to the concept of Internal Event, and composition can be specified using nesting relation. Also,

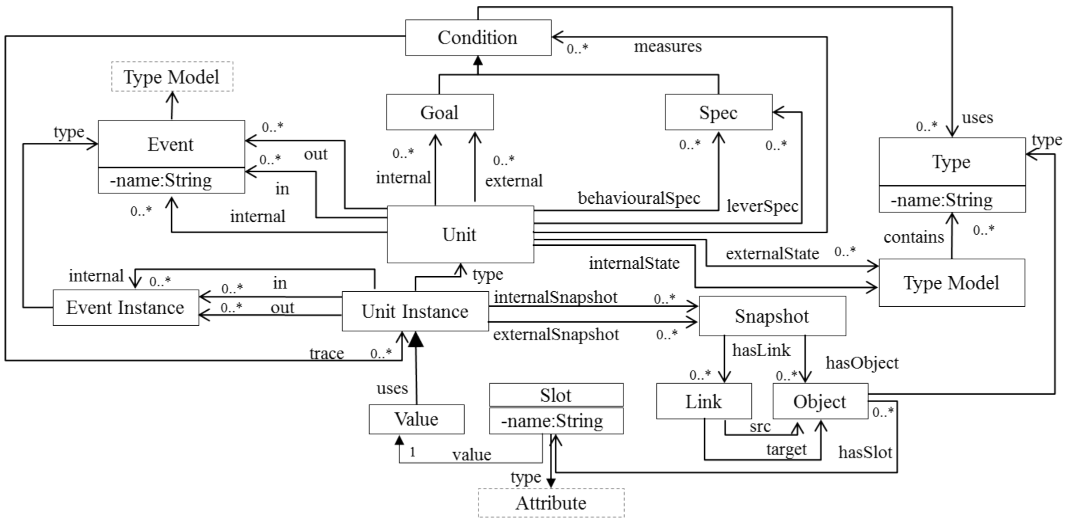


Figure 3: Implementation Meta Model.

Unit is adaptable as it can construct and reconstruct its structure; modular as it encapsulates the structure and behaviour of an organisation; intentional as it has its own goals; and compositional as it can be an assembly of Units.

We draw from a set of existing concepts to come up with the unit abstraction. Modularization and reflective unit hierarchy are taken from fractal component models (Barros et al., 2009). Goal-directed reactive and autonomous behaviour can be traced to agent behaviour (Bonabeau, 2002). Defining states in terms of a type model is borrowed from UML[3]. An event driven architecture (Michelson, 2006) supports flexible interactions between components, and the concept of intentional modelling (Yu et al., 2006) is adopted to enable specification of component goals.

3.3 Implementation

Fig. 3 describes the specialisation of the conceptual model of Fig. 2 for implementation and simulation semantic purposes. It can be read as follows. Organisation is a Unit that comprises a set of Units and strives to accomplish its stated Goal. It does so by responding to Events taking place in its environment (In-Events), processing them (as specified in Spec), and by interacting with other external Units in terms of Events raised/responded (OutEvents). A Unit may choose not to expose all events to the external world (InternalEvents). Spec (associated with Unit through behavioralSpec) is a declarative specification of event processing logic i.e., behaviour of the Unit. Thus, looking outside-in, a Unit is a Goal-directed agent that receives events (InEvents), processes them, and raises events (OutEvents) to be processed by other Units. Also, Unit is a parameterized entity whose structure and behaviour can be altered through Levers. The lever specification is a Spec that connects Unit with leverSpec. Internally, a Unit has current and historic states that comprise the instances of Event, Unit, Associations and Attributes. The model provides two abstractions namely Snapshots and Value to encapsulate instances. A Unit may choose not to expose the entire state to the external world (InternalState). A Unit interacts with other Units in a-priori well-defined Role-playing manner. TypeModel provides a type system for structural as well as behavioural aspects of a Unit. A language defined using the metamodel from Fig. 3, is termed as *ESL (Enterprise Simulation Language)* and has a meaning that is defined with respect to trace semantics expressed as a sequence of *Snapshots*. Sequence of snapshots describes the history and current state (i.e. D) of the organisation (O). Each unit has a specification that describes the organisational behaviour (P) and a unit conforms to a structure (S). A behaveioural specification (Spec associated with behaviouralSpec) is a predicate over traces that must be true for the projection of the overall organisation-trace that relates to the appropriate unit. Each unit has a goal (G) that governs its intent and behaviour. The semantics of goals are predicates over state traces or snapshots (i.e. D). Selected snapshots and slots can be marked as measure (M) for quantitative measurements. Unlike specifications, goals need not be true for every legal behavioural-trace: a goal may fail and it is the job of each individual unit to perform actions in order to achieve its goal. At a macro-level the entire organisation is defined as a unit whose goal must be measured by the simulation. At a micro-level individual units are agents whose goals govern

[3]http://www.omg.org/spec/UML/

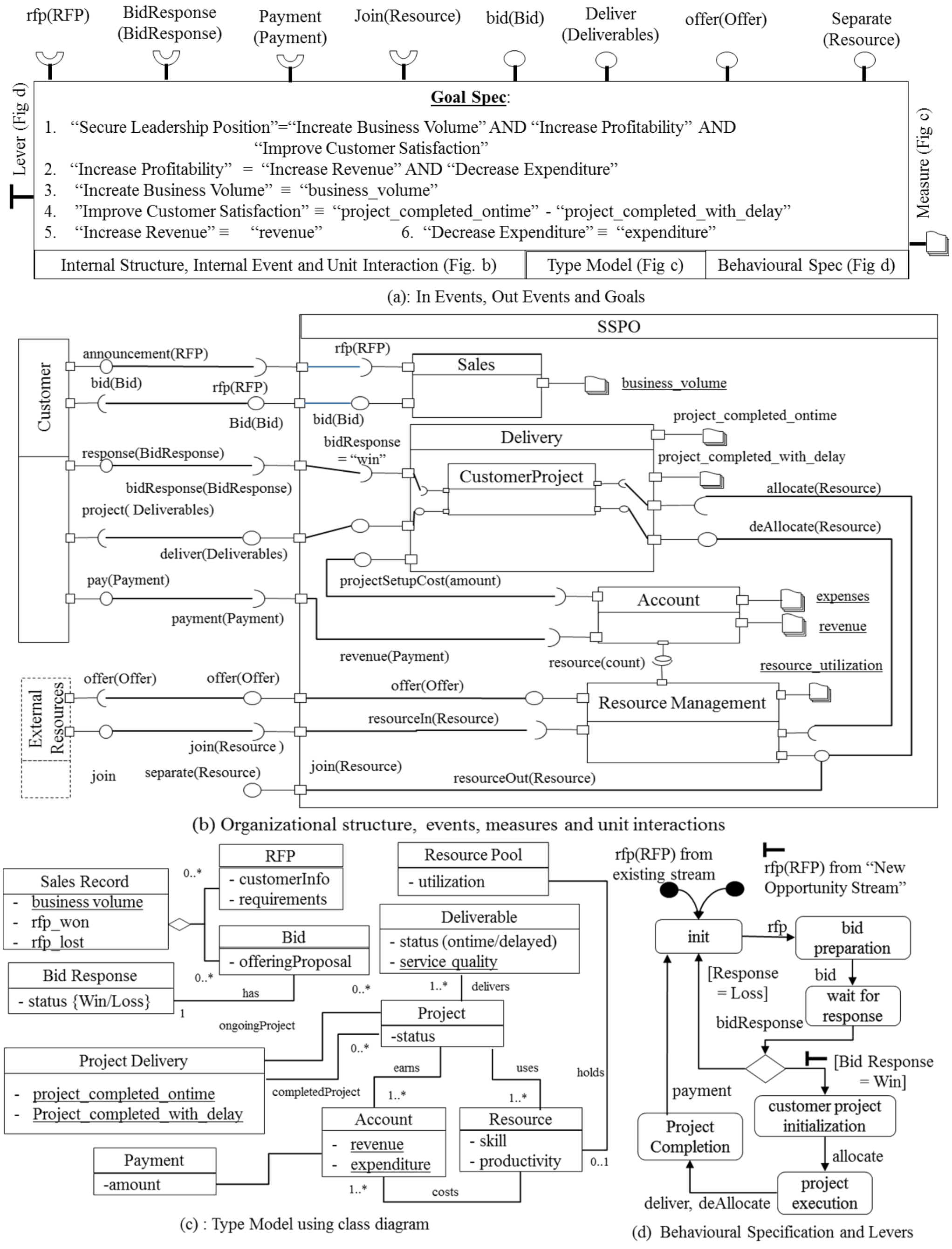


Figure 4: Models of Software Service Provisioning Organisation (SSPO).

their local intent. In a simulation scenario, the lever $l \in L$ can be selected though appropriate parameter value to lever Specs or modifying lever Specs. The observation of measures (M) is evaluation of appropriate snapshot values. Thus the scenario playing formulation i.e., $M = P_{<t0-t1>}(D_{t0}, L_{select}(O))$, is essentially setting initial simulation value (D_{t0}), selecting L_{select} from possible levers L, executing BAU process P for duration t1-t0 and evaluating of appropriate snapshot values. ESL was prototyped by extending an existing event-driven language with the concepts borrowed from actor model of computation (Hewitt, 2010), multi-agent systems (Van Harmelen et al., 2008), goals (Yu et al., 2006) and linear temporal logic (Pnueli, 1977). These concepts and their augmentation with conventional class models and temporal logic closely match the required features specified in Fig. 3. The ESL and simulation engine for ESL is implemented using DrRacket[4]. The implemented machinery is capable of specifying the relevant aspects of organisation and carry out if-what and what-if simulations.

4 ILLUSTRATIVE EXAMPLE

In this section we evaluate our approach by presenting a modelling and sample decision-making scenario of a software service-provisioning organisation. We consider an organisation that earns revenue by developing bespoke software for its customers. The organisation bids for various software projects in response to request for proposals (RFPs). Once a bid is won, the organisation initiates and executes projects using tried-and-tested process. This business as usual (BAU) scenario of the organisation is driven by high level goals of securing leadership position in terms of business volume, profitability and customer satisfaction.

An organisation performs BAU behaviour while achieving goals and explores several possibilities in case the goals are not achieved or are not achievable. There are many possibilities. For instance, improving operational efficiency while keeping organisation structure as well as operational processes unchanged. This strategy can be implemented by: Increasing number and skill-level of resources; Obtaining a predictive handle on demand; Reducing resource attrition, Reducing all sorts of delays such as recruitment, training, relocation etc; and any combination of these. On the other hand, some means of achieving goals might be more disruptive as they introduce changes in organisation structure and/or operational processes. For example, one can think of using generative techniques of software development instead of manual code-centric development thus necessitating major change in software development as well as project execution processes. The implementation of this case study example is detailed and can be seen to approximating to real life. The detail has considered various kinds of projects, different execution strategies and resource categorization derived from industry. But in the interest of size, here we consider a part of the case-study by limiting to a simple project classification and a relatively non-disruptive strategy for illustration purposes. Hence, the environment is characterised using a representative classification with customers offering four different kinds of projects – *High Margin High Risk (HMHR)* project, *Low Margin Low Risk (LMLR)* project, *Medium Margin High Risk (MMHR)* and *Medium Margin Low Risk (MMLR)*. The strategy is implemented using four levers namely “Increase Win Rate”, “New Opportunity Stream”, “Increase Resource Strength” and “Improve resource Skill”. Models and decision-making process for selecting the levers with best potential for achieving the desired organisational goal are illustrated below.

[4]http://racket-lang.org/

4.1 Models

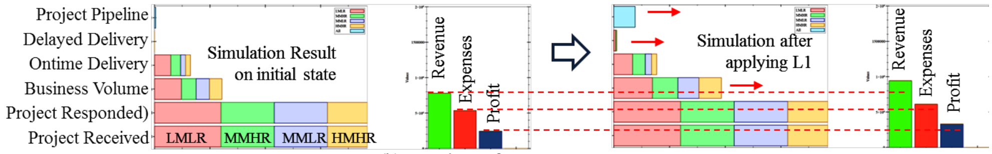
The model of the software service provisioning organisation is illustrated in Fig. 4. The organisation is visualized as a unit (SSPO unit) with four in-events, four out-events and an organisational goal as shown in Fig. 4 a. The key elements of the model are illustrated below:

Goal specification: SSPO unit targets a primary goal namely “Securing Leadership Position”. This goal is decomposed into three sub-goals namely “Increase Business Volume”, “Increase Profitability” and “Improve Customer Satisfaction” to support better qualitative and quantitative measurements. The “Increase Profitability” sub-goal is further decomposed into two sub-goals namely “Increase Revenue” and “Reduce Expenditure”. These goals and subgoals are described using predicates where terms are finally associated with TypeModel shown in Fig. 4 c. For instance, “Increase Business Volume” is associated with “business_volume” attribute of “Sales Record” class, “Improve Customer satisfaction” is associated with two attributes of “Project Delivery” class - “project_completed_ontime” and “project_completed_with_delay” (where these two attributes contribute positively and negatively respectively towards “Improve Customer satisfaction”), “Increase Revenue” is associated with “revenue” of “Account” class and “Reduce Expenditure” is associated with “expenditure” of Account class of the TypeModel.

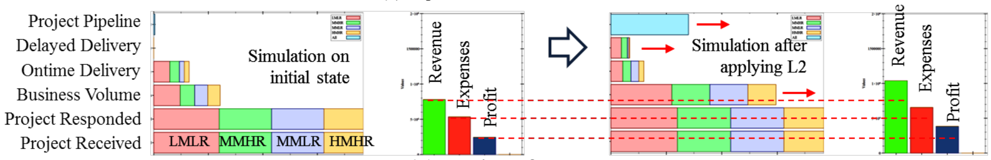
In and Out Events: The SSPO unit interacts with environment by receiving “rfp(RFP)” event, “bidResponse(BidResponse)” event, and “payment(Payment)” event from environment (in particular from customers), and responding “bid(Bid)” event and “deliver(Deliverable)” event to the customers. It also receives “join(Resource)” event from various sources, sends “offers(Offer)” event to the Resources who are outside of SSPO organisation, and send Resources to the environ-ment using “separates(Resources)” event for resign, termination and retirement. Events use TypeModel for specifying

Levers		Goal, Sub-goals and other Measures (symbols :- ↑ - increase, ↓ - decrease, ↕ - unknown, □ - eventually, Δ - delta)					Organization Goal
		Business Volume	*Revenue*	*Expense*	*Profitability*	*Customer Satisfaction*	*Secure Leadership Position*
L1	Increase Win Rate	↑	(Δ)↑	(Δ)↑	(Δ)↑	(□)↓	No Conclusion
L2	New Opportunity Stream	↑	(□)↑	↑	(□)↑	(□)↓	No Conclusion
L3	Increase Resource Strength	–	–	↑	↓	↑	No Conclusion
L4	Improve Resource Skill	–	–	(Δ)↑	(Δ)↓	(Δ)↑	No Conclusion
L5	L1 + L2 + L3	↑	↑	↑	↑	↑	↑

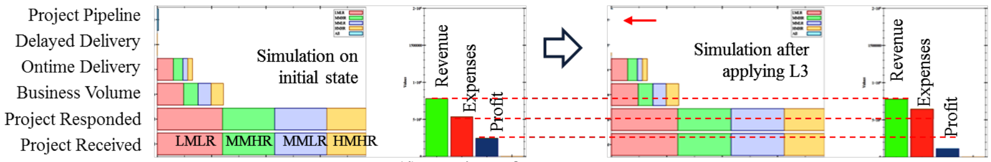
(a) Decision Table



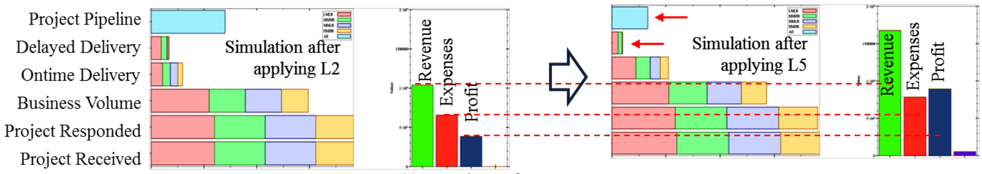
(b) Experiment for Row 1



(c) Experiment for Row 2



(d) Experiment for Row 3



(e) Experiment for Row 5

Figure 5: Decision Making and Simulation Results.

their parameters.

Internal events and organisation structure: Internal units, internal events and their interactions are depicted using component model in Fig. 4 b. The figure shows four sub-units namely “Sales”, “Delivery”, “Account” and “Resource Management” units with their in and out events. Interactions between SSPO and sub-unit, and between sub-units are also illustrated. For example “allocation(Resource)” and “deAllocation(Resource)” events are the interactions between “Delivery” unit and “Resource Management” unit whereas the delegation of event “rfp(RFP)” is the interaction be-tween SSPO and sub-unit. The interaction and structure can be static or dynamic. For example the “CustomerProject” is a unit is created once a Bid is won by SSPO unit and it resolves after producing “deliver(Deliverable)” event.

Behaviour: a simplified behaviour of SSPO unit is depicted using state diagrams in Fig. 4 d. The behaviour shows the transformation and life-cycle RFP. SSPO unit receives “RFP” through “rfp(RFP)” event; it then delegates to “Sales” unit; “Sales” unit works on “RFP” and transforms it to “Bid”; the “Bid” is transformed into a “CustomerProject” when a bid is won by SSPO (the intimation receives through “bidResponse(Bidresponse)”); internally the “CustomerProject” goes through many states and finally dissolves by responding event “deliver(Deliverables)” and deallocating resource using “deAllocate(Resource)”.

Measures: measures are state variables or condition over state variables. The attributes which are used in measures are highlighted in TypeModel of Fig. 4 c. The measures within SSPO unit are shown in Fig. 4 b. We consider 6 measures for SSPO unit - “Business Volume”, “Revenue”, “Expense”, “Profitability” and “Customer Satisfaction”. “Business Volume” measure represents the slot value “business_volume” attribute of “Sales Record” class, “Revenue” measure represents the slot value of “revenue” attribute of “Account” class, and “Expenditure“ measure represents the slot value of “expenditure“ attribute of “Account” class of TypeModel shown in Fig. 4 c. Similarly, the “Profitability” represents the conditions on the slot values of “revenue” and “expenditure” attributes of “Account” class, and “Customer Satisfaction” is for representing the condition on the slot values of “project_completed_ontime” and “project_completed_with_delay” attributes of “Project Delivery” class.

Levers: the leavers are the condition over events and its parameters in the context of behaviour. In this example, we consider 4 basic levers “Increase Win Rate”, “New Opportunity Stream”, “Increase Resource Strength” and “Improve Resource Skill”.

4.2 Decision-making

The decision-making process is about finding possible levers (L), evaluating them with respect to organisational goals and sub-goals (G), and selecting a set of levers ($L_{select} \in L$) that have the best potential to achieve the goal G. Simulation-based what-if and if-what scenario playing forms a cornerstone of this process. A high-level model of the system is created that encodes influences of various levers on [sub-] goals and of [sub-]goals on goals in a manner that is amenable to qualitative as well as quantitative decision making. Fig. 5 a shows simulation needs in a consolidated form (upper portion) and the rest of the sub-figures in Fig. 5 represent the simulation outputs from DrRacket based ESL prototype. Goal and sub-goals G form columns of the table (depicted in Fig. 5 a) with levers L forming the rows. Each cell of the table represents a what-if scenario for a lever $l \in L$ on goal $g \in G$ where the impact of a lever on a goal needs to be computed using simulation run. For example, first row in the table corresponds to “Increase Win Rate” lever. As can be seen, this lever has a positive impact on “Business Volume” sub-goal, marginally positive impact on “Revenue”, “Expense” and “Profitability” sub-goals, and eventual negative impact on “Customer Satisfaction” sub-goal. As a result, nothing conclusive can be said about impact of “Increase Win Rate” lever on the overall goal of “Secure Leadership Position”. The left half of figure Fig. 5 b depicts initial state of the organisation in terms of RFPs received, RFPs responded, RFPs won (i.e., “Business Volume”), On-time delivery, Delayed delivery, Project execution pipeline build-up etc without applying any levers. Significant points to be noted are: all projects are delivered on time, and there is no project execution pipeline build-up. Right half of Fig. 5 b depicts details of simulations carried out to determine the impact of “Increase Win Rate” lever on various sub-goals. As can be seen, “Business Volume” increases by about 30% but there is significant increase in the number of projects delivered with a delay some of which leads to penalties. As a result, profits do not increase in the same proportion as increase in “Business Volume”. Also, build-up in project execution pipeline is a concern that can lead to customer dissatisfaction that can potentially impact overall goal adversely. Fig. 5 c and Fig. 5 d. depict impact of levers “New Opportunity Stream” and “Increase Resource Strength” on the various sub-goals. Comparison of figures Fig. 5 b. and Fig. 5 c. shows the “Profitability” of “New Opportunity Stream” is much higher than the “Profitability” of “Increase Win Rate” however the factors associated with negative “Customer Satisfaction” are also high. On other hand, “Increase Resource Strength” shows positive impact on “Customer Satisfaction” but with an additional cost that brings down “Profitability”. Thus, as can be seen from the first four rows of figure Fig. 5 a, no lever individually can ensure the overall goal of “Secure Leadership Position” can be achieved. As a result, one has to explore what impact a combination of these levers can have. For example one can evaluate the combination of levers “Increase Win Rate” and “Increase Resource Strength” or levers “New Opportunity Stream” and “Increase Resource Strength”. Fig. 5 e. shows impact of levers “Increase Win Rate”, “New Opportunity Stream” and “Increase Resource Strength” applied together. As can be seen from Fig. 5 a, this conclusively leads to achievement of the overall goal. Further simulation can be done to fine tune the options (deciding quantitative figures) such as how much increase in resource strength is optimum when increase in win rate and the rfp from new opportunity stream is either known or can be estimated.

5 CONCLUSION

Organisational decision-making practice today relies excessively on human expertise. This is primarily due to unavailability of suitable technology support. Available technology support is found wanting either in completeness of specification of all relevant aspects of decision-making or in analysis rigour or both. We illustrated the limitations of state-of-the-art and state-of-the-practice of enterprise modelling and analysis techniques for decision-making of socio-technical organisation. As part of a pragmatic approach to organisational decision-making, this paper has presented a conceptual model, the accompanying implementation model that forms the basis of a high-level language and its simulation semantics.

The approach has been illustrated with a substantive example from the software services domain. We have shown the example can be modeled and simulated leading to the ability to influence the strategically selected measures. However, we recognise that the current implementation model (Fig. 3) of ESL is not sufficiently high-level for direct adoption by decision-makers. Our immediate next step is to develop high-level abstractions to support the core concepts of Fig. 3 in a business facing manner. In doing so, we will adopt language processing and model transformation technology to enable support for defining domain specific languages geared for specific problems. We note that decision-making is more a satisfaction problem rather than an optimisation problem. Consequently, we will draw upon game theory and computational economics to consider extending our proposed solution to impart this characteristic.

REFERENCES

Barjis, J. (2008). Enterprise, organization, modeling, simulation: Putting pieces together. *CEUR Workshop Proceedings*, 338:1–5.

Barn, B., Clark, T., and Kulkarni, V. (2014). Next generation enterprise modelling: The role of organizational theory and multi-agent systems. *ICSOFT-EA 2014*, pages 482–487.

Barros, T., Ameur-Boulifa, R., Cansado, A., Henrio, L., and Madelaine, E. (2009). Behavioural models for distributed fractal components. *Annales des Telecommunications/Annals of Telecommunications*, 64(1-2):25–43.

Bernus, P. and Schmidt, G. (2006). Architectures of information systems. In *Handbook on Architectures of Information Systems*, pages 1–9. Springer.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(SUPPL. 3):7280–7287.

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

Conrath, D. W. (1967). Organizational decision making behavior under varying conditions of uncertainty. *Management Science*, 13(8):B–487.

Dietz, J. (2001). Demo: Towards a discipline of organization engineering. *European Journal of Operational Research*, 128(2):351–363.

Fox, M. S. (1994). Issues in enterprise modelling. In *Information and collaboration models of integration*, pages 219–234. Springer.

Frank, U. (2002). Multi-perspective enterprise modeling (memo) conceptual framework and modeling languages. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 1258–1267. IEEE.

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

Iacob, M., Jonkers, D. H., Lankhorst, M., Proper, E., and Quartel, D. D. (2012). Archimate 2.0 specification: The open group. Van Haren Publishing.

Krogstie, J. (2008). Using eeml for combined goal and process oriented modeling: A case study. *CEUR Workshop Proceedings*, 337:112–129.

Locke, E. (2011). *Handbook of principles of organizational behavior: Indispensable knowledge for evidence-based management*. John Wiley & Sons.

McDermott, T., Rouse, W., Goodman, S., and Loper, M. (2013). Multi-level modeling of complex socio-technical systems. *Procedia Computer Science*, 16:1132–1141.

Meadows, D. H. and Wright, D. (2008). *Thinking in systems: A primer*. Chelsea Green Publishing.

Michelson, B. M. (2006). Event-driven architecture overview. *Patricia Seybold Group*, 2.

MOD (2005). Mod architectural framework technical handbook.

OMG (2011). Business Process Model and Notation (BPMN), Version 2.0.

OMG (2015). Business Motivation Model (BMM) Version 1.3.

Parsons, T. and Jones, I. (1960). *Structure and process in modern societies*, volume 3. Free Press New York.

Pnueli, A. (1977). The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57. IEEE.

Scheer, A.-W. and Nüttgens, M. (2000). *ARIS architecture and reference models for business process management*. Springer.

Shapira, Z. (2002). *Organizational decision making*. Cambridge University Press.

Van Harmelen, F., Lifschitz, V., and Porter, B. (2008). *Handbook of knowledge representation*, volume 1. Elsevier.

Vernadat, F. (2002). Ueml: Towards a unified enterprise modelling language. *International Journal of Production Research*, 40(17):4309–4321.

Wainer, G. A. (2009). *Discrete-event modeling and simulation: a practitioner's approach*. CRC Press.

Wisnosky, D. E. and Vogel, J. (2004). Dodaf wizdom: A practical guide to planning. *Managing and Executing Projects to Build Enterprise Architectures Using the Department of Defense Architecture Framework (DoDAF)*.

Yu, E., Strohmaier, M., and Deng, X. (2006). Exploring intentional modeling and analysis for enterprise architecture. *10th IEEE International Enterprise Distributed Object Computing Conference Workshops*.

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