A Model Based Approach for Complex Dynamic Decision-Making

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

[1] Tata Consultancy Services Research, Pune, India
[2] Sheffield Hallam University, Sheffield, United Kingdom
[3] Middlesex University, London, United Kingdom
{souvik.barat, vinay.vkulkarni}@tcs.com, t.clark@shu.ac.uk, b.barn@mdx.ac.uk

Abstract. Current state-of-the-practice and state-of-the-art of decision-making aids are inadequate for modern organisations that deal with significant uncertainty and business dynamism. This paper highlights the limitations of prevalent decision-making aids and proposes a model-based approach that advances the modelling abstraction and analysis machinery for complex dynamic decision-making. In particular, this paper proposes a meta-model to comprehensively represent organisation, establishes the relevance of model-based simulation technique as analysis means, introduces the advancements over actor technology to address analysis needs, and proposes a method to utilise proposed modelling abstraction, analysis technique, and analysis machinery in an effective and convenient manner. The proposed approach is illustrated using a near real-life case-study from a business process outsourcing organisation.

Keywords: Organisational decision making, Simulation, Model based approach, Conceptual model, Domain specific language.

1 Introduction

Modern organisations constantly rely on decision-making to select suitable courses of action that help in achieving their goals [1]. An effective organisational decision-making calls for precise understanding of various aspects of organisation such as goals, organisational structure, operational processes and the historical data describing operational details along with execution log. The inherent characteristics of modern organisations that include the socio-technical characteristics [2], complex and dynamic organisational structure [3], significant uncertainty [4], and emergent behaviour [5] make the decision-making a complex endeavor *i.e.*, complex dynamic decision making (CDDM).

We posit that effective CDDM hinges on the availability of: (i) information required for decision-making in a structured and machine-interpretable form, (ii) suitable machineries to interpret the information, and (iii) a method to help identify the relevant information, capture it in model form, and perform *what-if* analyses. The current practice of organisational decision-making that relies heavily on human experts typically working with primitive tools such as spreadsheets, word processors, and diagram editors *etc*. fares poorly on all the three criteria [6].

A wide range of Enterprise Modelling (EM) techniques, such as ArchiMate [7], IEM [8], MEMO [9], i* [10], BPMN [11], and System Dynamics (SD) [12], capture information of interest in a structured and/or machine interpretable form. They also support varying degree of analyses capabilities on a range of organisational aspects. However, they are found to be insufficient for CDDM [13, 14]. The actor languages and frameworks such as Kilim [15], Scala Actors [16], and Akka [17], in contrast, adopt the *actor model of computation* [18] to specify socio-technical characteristics. However, they are inadequate to express complex goal structure, organisational hierarchies, and behavioural uncertainty [13].

Therefore, it can be said that existing technological support can at best partly meet only two of the three requirements of effective CDDM *i.e.*, (i) the ability to conveniently capture the organisational goals, structure, behaviour, and their inherent characteristics and (ii) the ability to perform required analyses on available information. However, little is reported on how to use the relevant existing technologies, such as EM technologies and actor technologies, in a systematic manner for effective CDDM.

This paper presents a model-driven approach to capture necessary aspects of an organisation, such as goal, structure, and behaviour, along with their inherent characteristics, such as socio-technical characteristics and uncertainty, in a relatable and machine interpretable form and perform various *what-if* analyses leading to evidence-driven CDDM. In particular, this paper hypothesises that model-based simulation approach is an effective means to address CDDM and claims four contributions: i) a conceptual meta-model that represents necessary and sufficient aspects of the organisation along with the inherent characteristics of CDDM, ii) a simulation model that refines conceptual model for specific decision-making context, (iii) a pragmatic human-assisted technique to ascertain model validity, and (iv) a method to construct purposive simulatable models leading to *what-if* analyses for CDDM in a systematic manner.

The proposed conceptual meta-model caters to specification of *why*, *what*, *how*, *who*, *where* and *when* aspects [19], socio-technical characteristics as advocated in actor model of computation [18], and uncertainty [20]. The simulatable model advances the state-of-the-art actor technology [15, 16, 17] by supporting the notion of uncertainty and "time". The proposed method refines the management view of decision-making advocated by Richard Daft [3] while extending the modelling and model validation method advocated by Robert Sargent [21] so as to realize a simulation based approach to CDDM.

The paper is organized as follows. Section 2 provides background by highlighting necessary tenets of CDDM and reporting brief overview of existing EM techniques and actor technologies. It also summarises notable gaps restricting adoption of EM techniques and actor technologies for CDDM. Section 3 presents model-driven simulation-based approach to CDDM. The approach is illustrated in section 4 using a case study from business process outsourcing (BPO) domain. Section 5 discusses evaluation of the approach. The paper concludes with future work.

2 Background

This section presents the key requirements for affective CDDM and evaluates the state-of-the-art techniques and technologies with respect to these requirements.

2.1 CDDM Structure and requirements

Decision-making is a continuous and indispensable activity for all organisations. It requires deep understanding of various aspects of an organisation. Zachman Framework [19] recommends six interrogative aspects namely *why*, *what*, *how*, *when*, *where*, and *who* as necessary and sufficient information to precisely understand an enterprise. Conforming to Zachman Framework, we visualize an organisation using a set of concepts as shown in the class diagram in Figure 1 [22]. An *Organisation* has objectives or *Goals*, *i.e.*, *Why* aspect, that it aims to achieve. A Goal is typically assessed by evaluating a set of performance indicators or *Measures* that are indicative of organisational effectiveness along several dimensions such as time to market, growth rate, customer satisfaction, employee happiness index, entry into new areas *etc*. Organisational effectiveness in an *Environment* (*i.e.*, *where* aspect) is largely a function of its *Structure* (*i.e.*, *What* and *Who* aspects) and *Behaviour* (*How* and *When* aspects). Behaviour induces *State* changes thus producing *Trace* (i.e., historical record of States) over a period of time. A *Lever* represents a possible course of action available to organisation. Typically, applying a lever results in modification of either operational parameters or Goal or Behaviour or any combination of the three thus leading to modifications to the Trace. Thus, decision-making is a loop involving evaluation of possible Levers so as to identify the most promising one – untill the stated goal is achieved.

The conceptual structure of Figure 1 though necessary is not sufficient for effective CDDM. The system of systems structure of an organisation means the decision making problem can be positioned at various levels of granularity spanning from mega to macro to micro. This places additional demands of modularity and compositionality on the specification. As each [sub] system has own goals and the necessary wherewithal of achieving them, the specification needs to be capable of supporting *intentionality* and *autonomy*. As each of these [sub] systems operate over protracted time adapting constantly by responding to events taking place in their operating environments, the specification needs to be capable of supporting *reactive*, *temporal* and *adaptive* characteristics. Moreover, the specification must be capable of capturing the inherent *uncertainty*. Such a specification language along with its simulation engine seems necessary and sufficient infrastructure to support an iterative

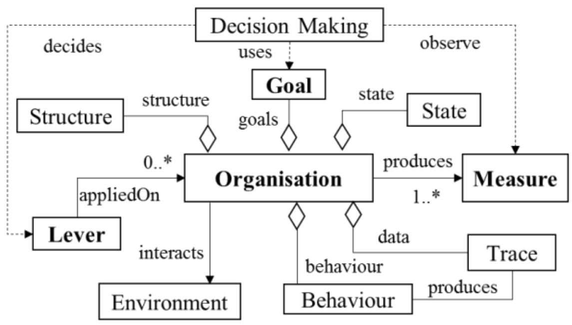


Figure 1 : Schema describing decision making concepts

Table 1. Requirements of CDDM

	Requirement	Description
Aspect	Why	Goals, objectives and intentions of multiple stakeholders
	What	Structural Specification with complex hierarchy and interactions
	How	Behavioural specification with interactions
	Who	Stakeholders and human actors of the system
	Where	Information about location
	When	Temporality in behaviour and adaptation
Socio-technical Characteristics	Modularity	A system can be decomposed into multiple parts.
	Compositional	Multiple parts should be composed to a consistent whole.
	Reactive	Must respond appropriately to its environment
	Autonomous	Possible to produce output without any external stimulus.
	Intentional	Intent defines the behaviour
	Adaptive	Adapt itself based on context and situation
	Uncertain	Precise intention and behaviour are not known a-priori.
	Temporal	Indefinite time-delay between an action and its response
DC	Measure	Ability to specify what needs to be measured
	Lever	Ability to specify possible courses of action
Analysis	Machine Interpretable	Models that are interpretable by machine (*i.e.,* support for simulation/execution)
	Top-down and Bottom-up	Support for top-down and bottom-up modelling and simulation to support reductionist view and emergentism

decision making loop wherein application of a *Lever* leads to modification of one or more *Measures* thus helping check whether a *Goal* (which is a sophisticated conditional expression over measures) is achieved or not [40, 41]. The list of requirements is summarised in Table 1.

From a methodology perspective, effective CDDM witnesses a curious dilemma. A system of systems structure involving autonomous [sub] systems indicates that organisation level *Goals* will be decomposed into various functional unit level Goals along the organisational *Structure* thus necessitating a top-down design approach. This implies that *Behaviour* of the organisation is known and hence specifiable. However, given the complexity of modern organisations and the inherent uncertainty, it is almost impossible to know the overall behaviour of organisation. The behaviour is typically known only for highly localized contexts i.e., functional units thus suggesting a bottom-up design approach wherein the overall organisation behaviour emerges from the behaviour of its interacting functional units. As a result, the specification language and analysis techniques need to be cognizant of top-down and bottom-up approach [23, 24] as described in Table 1. Also, effective CDDM calls for a method providing help with: (i) evaluating if the desired *Goal* is achieved, (ii) identifying the most appropriate *Lever* amongst many candidates, and (iii) applying the *Lever*.

2.2 Review of state of the art and practice

The state-of-the-art specification and analysis techniques approach the decision-making problem in two ways namely: data-centric approach and model-centric approach.

The data-centric approach makes use of sophisticated AI-based pattern recognition and predictive analysis techniques on relevant past data or *Trace* to predict future outcomes. This approach has worked well when *Trace* of an *Organisation* is comprehensive and the future is typically a linear extrapolation of the past. However, the two conditions are increasingly not being met for modern large enterprises thus leading to inappropriate decisions for emerging business context[1].

The model-centric approaches, in contrast, characterise the real organisation in the form of representative models which span across a wide spectrum. At one extreme of the spectrum are models that provide a well-defined structure for the organisational aspects of interest and rely on a variety of visualisation techniques to help humans obtain the desired understanding of the organisation. For instance, ArchiMate [7] is one such specification. At the other extreme of the spectrum are machine interpretable and/or simulatable specifications. They are capable of precise analyses for one or limited aspects. For instance, BPMN (Business Process Modelling and Notation) [11] analyses and simulates the behavioural aspect, i* [10] analyses the high level goals and objectives, and System Dynamic model simulates dynamic behaviour of the system. The multi-modelling and co-simulation environments, such as DEVS (Discrete EVent system Specifications) [25], AA4MM (Agent & Artifact for Multi-Modeling) [26], AnyLogic [28] and MEMO (Multi-perspective enterprise modeling) [9] technology, demonstrate further advancements by supporting the analysis of multiple aspects. Principally they adopt a top-down [23] approach to help analyse enterprises where the mechanistic world view holds. On the other hand, the languages and specifications advocating an actor model of computation [18] and agent-based systems [28] support emergentism [24] through bottom-up simulation. They fare better in analysis of systems comprising of adaptive and socio-technical elements.

Thus, the above mentioned techniques and technologies capture only a fragment of what ought to be captured and analysed for effective CDDM as illustrated in Table 1 [13]. In particular, the enterprise modeling languages are incapable of specifying uncertainty as well as emergent behaviour, and actor/agent languages are inadequate to conveniently express required characteristics such as the complex goal structure, organisational hierarchies, and behavioural uncertainty [22]. Moreover, EM specifications and actor based languages fall short as an intuitive and closer-to-the-problem specification as they are not designed for CDDM.

From a methodological viewpoint, the goal specification languages such as i* [10] and EKD [29] advocate a top-down method. EM languages such as ArchiMate, MEMO, and 4EM [30] advocate a top-down method and a globalized view of the system to represent the *Goal*, *Structure* and *Behaviour* of organisation in an integrated manner. BPMN [11] and SD model [12] predominantly support top-down approach and reductionist view of analyses [39]. On the other hand, actor languages and frameworks [15-17] advocate localised view, bottom-up approach, and emergentism. The reported methodological advancements also fail to support desired design principles. For example, DESIRE (DEsign Specification of Interacting REasoning components) [34] and MEMO based decision-making process [35] propose top-down model and reductionist

[1] https://hbr.org/2014/09/9-habits-that-lead-to-terrible-decisions

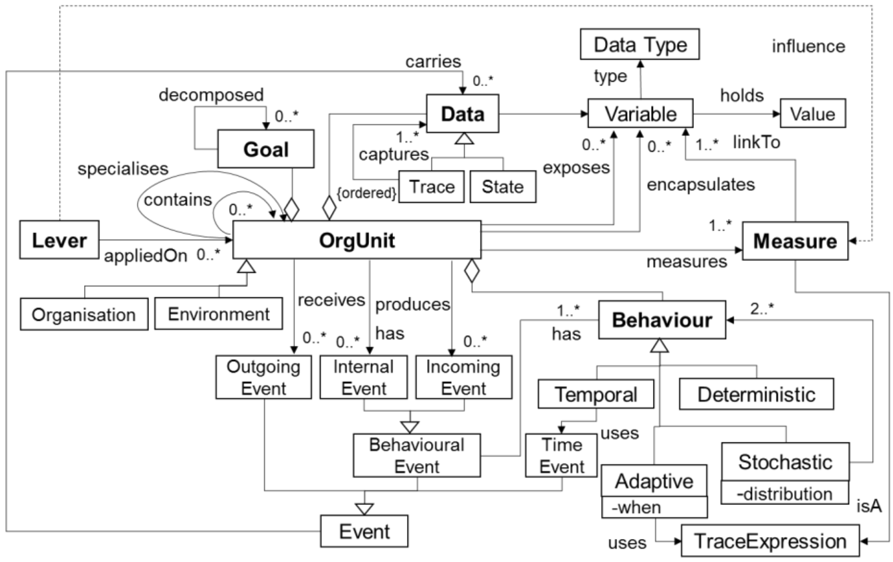


Figure 2: CMModel – a metamodel to represent organisation

what-if analysis. On the other hand, [36] advocates bottom-up approach using Belief-Desire-Intention (BDI) paradigm. Thus, there exists no single approach capable of combining top-down / bottom-up [23] design principle, reductionist / emergentism analysis techniques [24], and localized / globalized perspectives as desired. Moreover, the existing approaches are also found wanting in terms of ensuring model validity [21] and correlating with the management view of decision-making.

The next section describes our approach that addresses some of the essential specification limitations, overcomes inadequacy of analysis needs, and bridges the existing gap in methodical support.

3 Approach

Our approach to CDDM uses a model-based representation of organisation capable of supporting *what-if* simulation with a comprehensive design and analysis method providing the integration glue. In particular, we propose three artefacts that include: (i) a conceptual meta-model, termed as CMModel, to represent relevant aspects of an organisation along with the characteristics described in Table 1, (ii) a simulatable model, termed as ESLMModel, along with simulation machinery to support analyses needed for CDDM, and (iii) a method to help construct these models so as to perform *what-if* analyses leading to evidence-driven CDDM.

3.1 Conceptual Model

The CMModel meta-model is depicted in Figure 2. As shown in the Figure, the key abstraction of CMModel is *OrgUnit* that represents an autonomous self-contained functional unit having high internal coherence and low external coupling. Each OrgUnit has its own *Goal*, contains *Data*, deals with a set of interacting *Events*, and may have specific *Behaviour*. The Goal represents the intention or objective of an OrgUnit. A Goal can be decomposed into sub-Goals, sub-sub-Goals to represent hierarchical goal structure. Data captures the current *State* and sequence of historical states, *i.e.*, *Trace*, using a set of typed entity *Variables*. An OrgUnit may encapsulate and/or share Data by encapsulating and/or exposing Variables. OrgUnit responds to three kinds of Events namely *OutgoingEvent*, *BehaviouralEvent* and *TimeEvent*. The OutgoingEvents are triggered from an OrgUnit as part of its reactive behaviour. Each OutgoingEvent specifies the Data that it carries while reacting to an Event. The BehaviouralEvent specifies behaviour that is a response to an event and the Data it consumes. The BehaviouralEvent is further classified into two types namely *InternalEvent* and *IncomingEvent*. The IncomingEvents are consumed by OrgUnit, and the InternalEvents are the events that are internal to an OrgUnit. The TimeEvent is a special event that represents the concept of “Time” such as “Day”, “Month” or a “Year”.

The *Measure* and *Lever* of an OrgUnit represent the Measure that an OrgUnit owns and the Lever that are relevant for an OrgUnit. Essentially, a Measure can be represented using a set of Variables and the Lever describes the change specification of *Variables*, composition relationships, Behavioural specification and/or Goals. We visualise the notion of organisation and its environment as specialised OrgUnit namely *Organisation* and *Environment* as shown in Figure 2.

By the virtue of being composable, OrgUnit abstraction is capable of modelling the system of systems nature of modern organisation. The composability can be specified using *contains* relationship. The meta-model advocates four kinds of Behaviour namely *Deterministic*, *Stochastic*, *Temporal* and *Adaptive*. The Deterministic behaviour describes the behaviour which is known with certainty. Essentially, the *known known* kinds of behaviour [20] can be specified using Deterministic Behaviour. The Stochastic behaviour describes uncertain Behaviour or *known unknown* kind of behaviour [20]. We use probabilistic distribution to specify Stochastic Behaviour. The Temporal Behaviour describes the temporal delays in interaction pattern, and the Adaptive Behaviour describes adaptation rules by describing *what* will change *when*.

The proposed meta-model is grounded into a set of existing concepts. The modularisation and unit hierarchy are taken from the notion of component abstraction. The goal-directed reactive and autonomous behaviour can be traced to actor behaviour [18, 37]. Defining states in terms of a type model is borrowed from UML. An event driven architecture is introduced for reactive behaviour. The concept of intentional modelling [10] is adopted to enable specification of goals. The behavioural classification and uncertainty is defined from the notion of uncertainty defined by Donald Rumsfeld [20].

We argue that CMModel meta-model realises the structure defined in Figure 1 and satisfies the requirements stated in Table 1. Event definition, Data, and OrgUnit structure together specify the *what* aspect, OrgUnit help specify the *who* and *where* aspects,

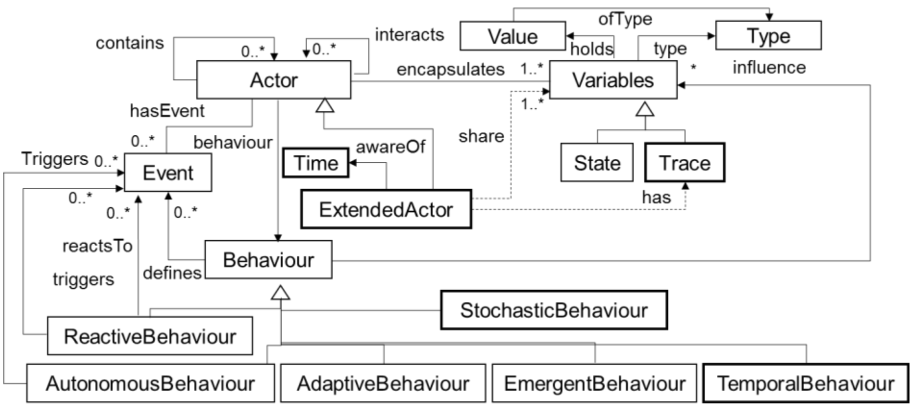


Figure 3 : ESL meta-model (ESLMModel)

Goal specification specifies the *why* aspect, and Behaviour specifies the *how* and *when* aspects. The concept of OrgUnit ensures modularity and encapsulation, the Event helps to specify reactive nature, InternalEvent and TimeEvent collectively specify the autonomous behaviour, Stochastic behaviour helps in specifying uncertainty, the Temporal behaviour and TimeEvent specify the temporal behaviour, and Adaptive behaviour is capable of specifying the adaptive nature of an OrgUnit. We argue that the *contain* relationship of OrgUnit and OrgUnit specific localised Behaviour definition help in bottom-up design, whereas the *contain* relationship of OrgUnit, Goal *decomposition* relationship, and an ability to share Variables using *exposes* relationship help in top-down design. The next section introduces a specification that has capability to represent the information captured using CMModel in a simulatable form.

3.2 Simulatable Model

We extend the notion of traditional *actor* definition [37] to specify enterprises. The adopted concepts from actor model of computation and proposed extensions are depicted using a meta-model, termed as ESLMModel, in Figure 3. The extended concepts are highlighted with bolded boxes and extended associations are represented using dotted lines. The *Enterprise Simulation Language* (ESL)[2] provides an implementation for ESLMModel.

As shown in Figure 3, the notion of traditional *Actor* encapsulates its *State*, has specific *Behaviour* and interacts with other Actors using a set of *Events*. The State of an Actor is defined using a set of typed *Variables* where each Variable holds *Value*. The Behaviour of an Actor principally represents four kinds of behavioural patterns namely reactive behaviour, autonomous behaviour, adaptive behaviour and emergent behaviour. ESLMModel represents supported behavioural patterns using four kinds of Behaviour namely *ReactiveBehaviour*, *AutonomousBehaviour*, *AdaptiveBehaviour* and *EmergentBehaviour*.

[2] https://www.gitbook.com/book/tonyclark/esl/details

Table 2: Conceptual mapping from CMModel to ESLMModel

CMModel	ESLMModel	CMModel	ESLMModel
OrgUnit	ExtendedActor	Trace	Actor Variable
Data	Actor Variables		
Goal	Expression over Actor Variables	Deterministic	DeterministicBehaviour
Event	Event	Stochastic	StochasticBehaviour
Measure	Expression over Actor Variables	Temporal	TemporalBehaviour
Lever	ESL specification	Adaptive	AdaptiveBehavioural

The ESL extends the notion of traditional Actor along four dimensions: (i) representation of historical state information or *Trace*, (ii) the notion of "Time", (iii) the notion of shared Variables that breaks pure encapsulation without compromising the correctness of state space of an actor, and (iv) the notion of uncertainty. The extensions (i), (ii) and (iii) are introduced using a specilised Actor entity named *ExtendedActor* and the extension (iv) is introduced as a specialised behavioural type named StochasticBehaviour in the ESLMModel (*see* Figure 3). The notion of "Time" helps specify temporal behaviour that we represent using a specialised Behaviour named *TemporalBehaviour* in ESLMModel.

ESL provides standard language constructs namely assignment, expression evaluation, loop, recursion, message passing, *etc.*, to express Deterministic Behaviour. Stochastic Behaviour is expressed using *'probably(p) x y'* construct that evaluates to *x* in p% of cases and otherwise to *y*. ReactiveBehaviour reacts to an Event or a set of Events, AutonomousBehaviour is typically triggered based on state Variables and/or Time, and AdaptiveBehaviour has a conditional expression over State and Trace Variables. The EmergentBehaviour, on the other hand, remains unspecified.

We propose a set of transformation rules to derive ESL specification from CMModel. The OrgUnit and its specialisation, *i.e., Organisation* and *Environment*, map onto ExtendedActor, interactions among OrgUnits map onto event specifications, and OrgUnit Variables map onto Variables of ExtendedActor. Measure maps onto Variables of ExtendedActors, Goal maps onto an expression over Variables of ExtendedActors, and the behavioural descriptions of OrgUnit map onto the behavioural specifications of ExtendedActors. The conceptual mapping from CMModel to ESLMModel is illustrated in Table 2. Next section describes a method to construct models using CMModel, transform the constructed model into ESL specification, and perform *what-if* analysis in a systematic manner.

3.3 Method

We propose an integrated and iterative method to effective CDDM that comprises of three essential activities: (i) construction of a simulatable model from available information of an organisation, (ii) ascertain model validity, and (iii) simulate model for *what-if* analyses leading to evidence-driven CDDM. The proposed method contains six steps namely *Define Decision Problem* [S1], *Conceptualisation of Organisation Model*

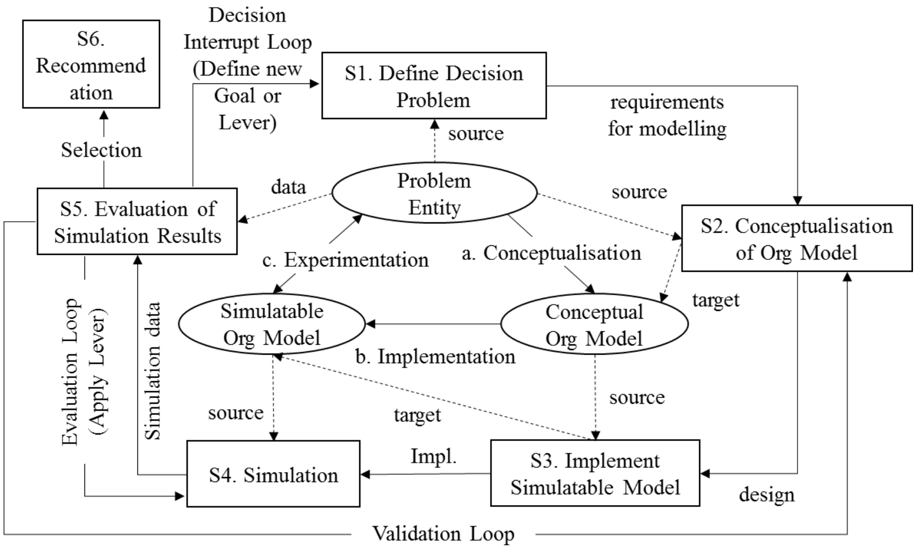


Figure 4: Overview of modelling and simulation method

[S2], *Implement Simulatable Model* [S3], *Simulation* [S4], *Evaluation of Simulation Results* [S5], and *Recommendation* [S6] as shown in Figure 4. Step S1 formalises the decision problem and defines the scope for *what-if* scenario playing by describing the Goals, Measures and Levers of an Organisation. Step S2 conceptualises a purposive model that represents a real system for decision problem defined in S1. Step S3 transforms the conceptual model into a simulatable model. Step S4 simulates the scenario defined in step S1. Step S5 evaluates the simulation results with step S6 providing recommendations.

Conceptually the proposed method realises the modelling and validation method proposed by Robert Sargent in [21] (henceforth referred as *M&V Method*) and adopts decision-making techniques recommended in management science [3]. From M&V Method, we adopt the notion of three representations namely *problem entity*, *conceptual model* and *computerized model*, and a two-step model construction process that includes *Conceptualisation* and *Implementation* steps to transform a real-life problem into valid analysis model as shown in Figure 4. We also adopt the *operational validity* [21] described in M&V Method to ascertain model validity. From management science, we adopt an iterative exploration of decision alternatives as recommended in [3] and the concept of decision interrupts [38] to explore decision alternatives that emerges while evaluating other decision alternatives.

In agreement with M&V Method, we consider the problem entity is the real organisation, the conceptual model is a purpose specific conceptual model that is necessary and sufficient to represent it for decision-making, and the computerised model is a machine interpretable equivalent of the conceptual model, *i.e.*, simulatable model. From a method perspective, the *Conceptualisation* step constructs a conceptual model from problem entity description (typically described in natural language), and *Implementation* step transforms the conceptual model into a simulatable model so as to use model-based simulation. The detailed activities of five method steps of Figure 4 are illustrated below:

Conceptualise Organisation Model [S1]: A decision problem typically starts with a high-level *Goal* or objective of an organisation. It should be possible to decompose a high-level Goal into sub-Goals, sub-sub-Goals etc., to the desired level of granularity. It should be possible to identify a set of variables that need to be observed in order to determine whether the finest-level goal is met or not, i.e., *Measures*. It should be possible to identify a set of course of actions or *Levers* that may influence the given set of Measures. The method step *Define decision problem* defines the Goals, Measures and Levers of an Organisation from problem entity description using three sub-steps namely *Goal Definition*, *Measure Identification* and *Lever Identification*.

The *Goal Definition* sub-step uses a top-down approach to define goals and goal decomposition structure. *Measure Identification* sub-step identifies *Measures* for all leaf-level Goals of constructed goal model. We use i* specification to visualise the goals of a decision problem. We represent Goals using the *Soft Goal* of i* notation, Measure using i* *Task* of i* notation, and *Goal-to-Measure* relationships using Task-Goal dependency relationship of i* notation [10].

The sub-step *Identify Levers* focuses on two activities: (i) identify a set of Levers that may impact identified Measures, and (iii) formulate a table, termed as *decision table*, by considering the identified Levers as rows and Measures as illustrated in Figure 7 in section 4.

Conceptualisation of Organisation Model [S2]: This step captures the *Structure*, *Behaviour*, *State* and *Trace* of an organisation and overlays the Goals, Measure and Levers identified in method step S1 using OrgUnit abstraction defined in CMModel (as depicted in Figure 2). Essentially this method step performs four activities namely (i) *Identify OrgUnits*, (ii) *Define OrgUnit*, (iii) *Define GM-L*, and (iv) *Specify Behaviour*. Activity *Identify OrgUnits* identifies prospective OrgUnits such as organisational units, sub-units, stakeholders, resources, and environment from problem entity. Activity *Define OrgUnit* forms OrgUnits by specifying *Variables* to represent State and Trace information, and the *Events* that help interacts with other OrgUnits. It also identifies containment relationship to describe composition and decomposition relationships of identified OrgUnits. In general, the activity *Identify OrgUnit* starts with organisation as an OrgUnit, and iterates over activity *Identify OrgUnit* and activity *Define OrgUnit* by navigating the decomposition and/or composition relationships. Essentially, it uses a middle-out approach that combines top-down and bottom-up design principles.

The activity *Define GM-L* identifies the Goals that an OrgUnit owns, the *Measures* that it can produce, and the Levers that can be applied on it. The activity *Specify Behaviour* captures the behavioural specification of identified OrgUnits.

Implement Simulation Model [S3]: This method step converts a Conceptual Organisation model defined using CMModel into machine interpretable specification, i.e., ESL specification. Essentially, S3 transforms all OrgUnits into ExtendedActors by applying transformation rules defined in Table 2.

Simulation [S4]: We use ESL based simulation to analyse *what-if* scenario formulated in method step S1. This step simulates the simulatable organisation model (with or without Lever), observes Measures from a simulation run, and captures results in a row of *decision table* formulated in method step S1.

Evaluation of Simulation Results [S5]: This step evaluates simulation results captured in decision table. Human expert interprets the simulation results triggering one of the following possibilities: (i) initiate a *Validation Loop* that iterates method steps S2-S3-S4-S5 in case simulation results of known scenario don't match the expected outcome (*i.e., operation validity* is not satisfied), (ii) explore next Lever of a decision table by triggering an *Evaluation Loop* that iterates method steps S5-S4-S5, (iii) select the best possible Lever once all levers are evaluated through simulation (*i.e.,* S5 to S6 transition), (iv) identify a new Lever *i.e.,* add a new entry in decision table and reiterate the overall method using *Decision Interrupt Loop* described in Figure 4.

Recommendation [S6]: This step recommends one or more Levers that can be implemented in real organisation.

3.4 Validation

Our method uses a validation loop that iterates over method steps S5-S2-S3-S4-S5 and compares experimental results with real or predicted data to ascertain model validity. We consider operational graphics [21], *i.e.,* graphical representation of *Measures* as a basis for evaluation, and rely on human experts to certify the validity. For model validation, we rely solely on operational validity through manual certification of simulation results of known scenarios. Other validation techniques, such as data validity or conceptual validity, while being effort and time intensive, provide no additional certainty as discussed in [21]. We next illustrate the proposed method using a real-life decision-making scenario.

4 Illustration

This section presents a problem entity from business process outsourcing (BPO) industry and illustrates the execution of proposed method along with their outcomes.

4.1 Problem Entity

In BPO, a class of organisations, termed as *customers*, outsource their business processes to another set of organisations, which is termed as *vendors*. Customers outsource their business process for a variety of reasons such as reducing *Cost* (C), increasing *Efficiency* (E), bringing about a major transformation, *i.e., Delight* (D). The vendors offer value-added services to their customers and earn revenues while servicing outsourced business processes. Considering the accruable business benefits of vendors, the outsourced business processes are classified into three broad buckets namely *Sunrise* (SR), *Steady* (ST) and *Sunset* (ST). The Transcript Entry process of Healthcare verticals

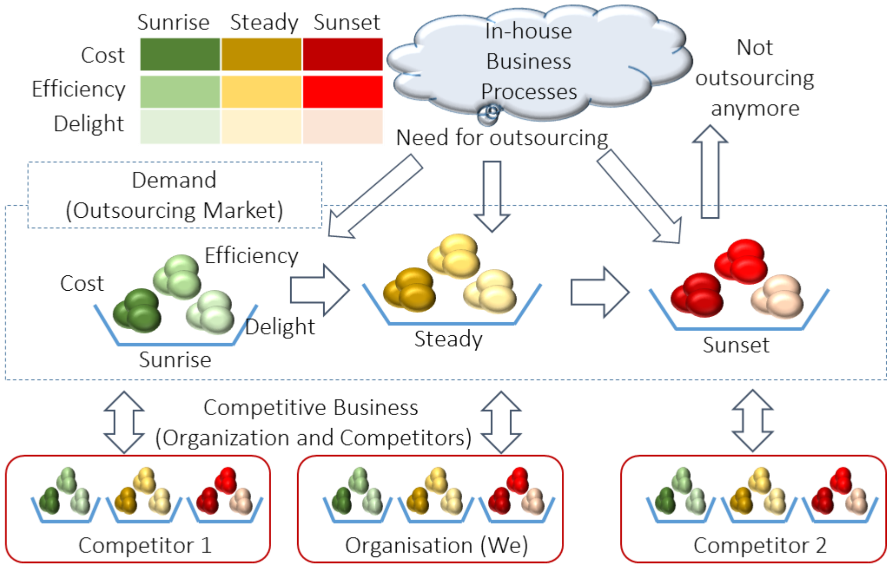


Figure 5 : Overview of Business Process Outsourcing Scenario

is one of the early adopters of BPO and has derived almost all potential benefits accruable from outsourcing (known as *Sunset*). On the other hand, IT Infrastructure Management process being a late adopter of BPO, has a large unrealized potential to be tapped (known as *Sunrise*). And there are processes such as Help Desk, Account Opening, Monthly Alerts *etc.*, that fall somewhere in between the two extremes as regards benefits accrued from BPO (known as *Steady*). Thus, the outsourced business processes of BPO industry can be described using a 3 x 3 matrix as depicted in Figure 5.

The business-as-usual (BAU) operational process of a BPO is largely limited to a set of interactions between customers and vendors. A *customer* publishes RFP (Request For Proposal) with an intension to outsource a business *process*. Interested *vendors* bid for RFP. Typically, factors such as *Quadrant* (i.e. ranking as per independent agency such as analysts), *FTE Count Range* (i.e. Full Time Employees to be deployed on the outsourced process), *Billing Rate Range* (i.e. per hour rate of FTE), *Organisation Size* (the number of employee) and *Track Record* (i.e., familiarity with the processes being outsourced), influence who wins the bid. The soft issues such as *Market Influence* (i.e. perception of the market as regards delivery certainty with acceptable quality), the rapport with the vendor *etc.*, also play a part in bid evaluation. In addition to these known factors there could be some uncertainty in bid evaluation criteria (in other words, bid evaluation criteria can't be fully known a-priori).

It is common observation that BPO outsourced business process engagements come up for renewal after few years (typically 3 to 5 years). A customer may renew the contract with the existing vendor on modified terms (typically advantageous to the customer) or may opt for rebidding. Factors influencing the renewal decision are reduction offered in *FTE Count*, *Billing Rate*, number and degree of escalations, perception that the external agent has as regards ability to meet the process engagement requirements, inherence uncertainty, etc. Contracts that fail to get renewed become candidates for later bidding. Figure 5 shows a high level schematic of the BAU of BPO industry. The interaction pattern between customer and vendor is depicted in Figure 6.

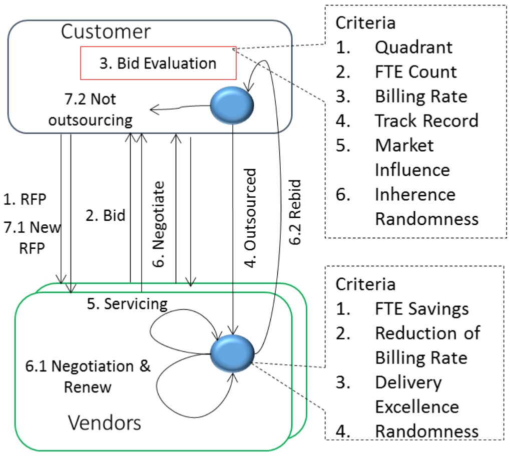


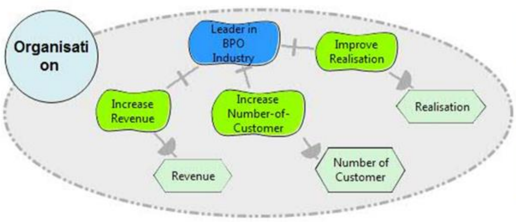
Figure 6 : Interactions and Behaviours

Given the above scope or a problem entity, the vendors mostly explore the decision-making problems that include: Will continuation with the current strategy keep “Me” viable ‘n’ years hence? What alternative strategies are available? How effective will a given strategy be? By when will a given strategy start showing positive impact? *Etc*.

In this paper, we consider a BPO vendor who would like be the leader in BPO industry with respect to the revenue, market share, and *realisation* (where the term *realisation* represents the revenue earned by each employee per hour). The next subsections describe the execution of method steps depicted in Figure 4 and their outcomes.

4.2 Define Decision Problem

The proposed method starts with a method step *Define Decision Problem* [S1] that formulates goal models and a decision table. We consider, a vendor, termed as “WE” vendor, aims to be the “Leader in BPO Industry”. The method step S1 decomposes “Leader in BPO Industry” Goal of “WE” vendor into three sub-Goals namely “Increase Revenue”, “Increase Number-of-Customer”, and “Improve Realisation”. It identifies three Measures namely “Revenue”, “Number of Customers”, and “Realisation” to assess three leaf-level Goals. The primary goal, goal decomposition structure and associated Measures are depicted in Figure 7(a).

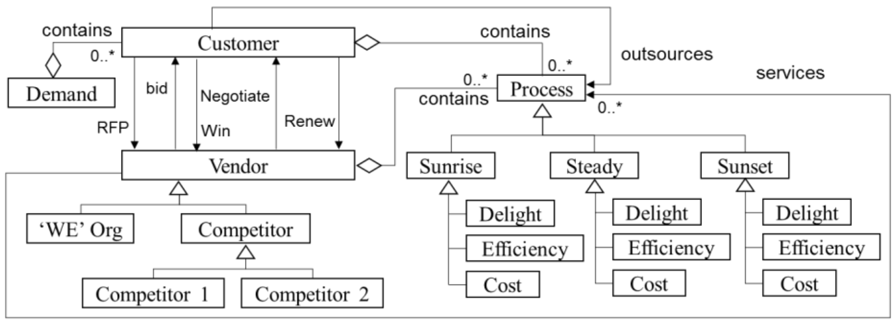


(a): Goal decomposition and Measures

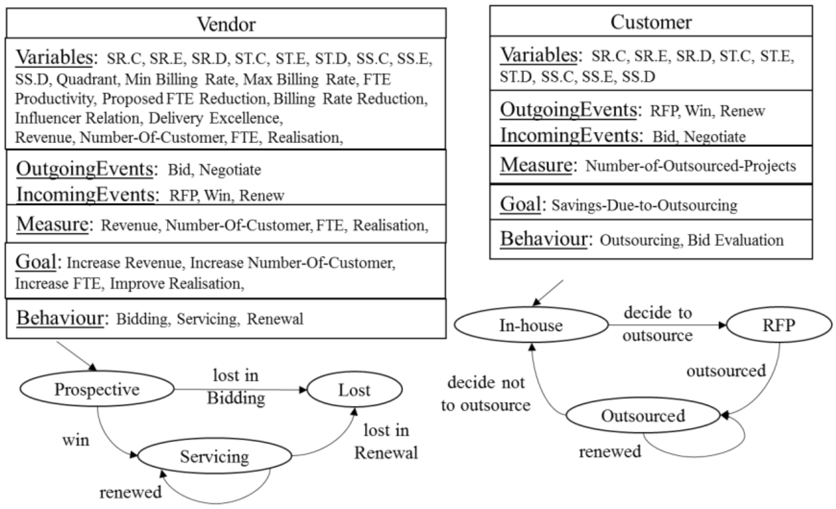
Lever	Revenue	Number of Customer	Realisation
No Lever			
Improve Skillset of Existing Employee			
Robotic Process Automation (RPA)			

(b): Decision Table

Figure 7: Output of method step Define Decision Problem



(a) Structure of of BPO Industry using CMModel



(b) Definition of Vendor and Customer OrgUnits

Figure 8: Conceptual Organisation Model of BPO Industry

The method step S1 also identifies possible Levers that may influence the Measures and thus Goals. In this paper, we consider two Levers namely "Improve skillset of existing employee" and "Introduce Robotic Process Automation (RPA)" as illustration. Identified Levers and Measures are shown in a form of decision table in Figure 7 (b).

4.3 Conceptualisation of Organisational Model

Method step S2 iteratively forms *Conceptual Organisation Model* from *problem entity* using four activities namely *Identify OrgUnit*, *Define OrgUnit*, *Define GM-L* and *Define Behaviour*. The activity *Identify OrgUnit* initially identifies three key OrgUnits namely "Customer", "Vendor", and "Process". The next activity *Define OrgUnit* captures structural relationships, Variables, and Event definitions of three OrgUnits. The Variable, IncomingEvent and OutgoingEvent of Vendor and Customer OrgUnits are illustrated in Figure 8 (b). Essentially the Vendors OrgUnit has a set of Variables to represent portfolio baskets (i.e., flattened out 3 x 3 matrix), the characteristics Variables such as *Quadrant*, *Min Billing Rate*, *Max Billing Rate*, *FTE Productivity*, *Proposed FTE Reduction* (during process engagement renewal time), *Proposed Billing Rate Reduction* (during project renewal time), *Influencer Relationship*, *Delivery Excellence* of the vendor OrgUnit. The OrgUnit also captures the state Variables that indicate Measure of Vendor OrgUnit such as *Revenue*, *Number-Of-Customer*, and *Realisation*.

The outcome of the iterative loop involving two activities namely *Identify OrgUnit* and *Define OrgUnit* is depicted using a class diagram in Figure 8 (a). As shown in the figure, several new OrgUnits are identified and elaborated over iterations. The “Process” OrgUnit is specialised into nine OrgUnits to represent business processes described using a 3 x 3 matrix of Figure 5. The Vendor is specialised into two entities namely “WE” vendor and “Competitor” vendor. The “WE” vendor represents a vendor under consideration, and the “Competitor” vendor represents the competitor vendors of “WE” vendor. There could be several competitors who adopt a range of strategies to compete in BPO industry. We consider two types of competitors namely “Competitor 1” and “Competitor 2” as shown in Figure 8 (a). The other relationships such as Customer “contains” various kinds of Processes, Vendor “outsources” Processes, Vendor “contains” a set of Processes and Vendor “services” Processes are defined in this method step. The interactions patterns between Customer and Vendors are also become explicit in this method step. The relationships and interaction patterns between OrgUnits are illustrated in Figure 8 (a).

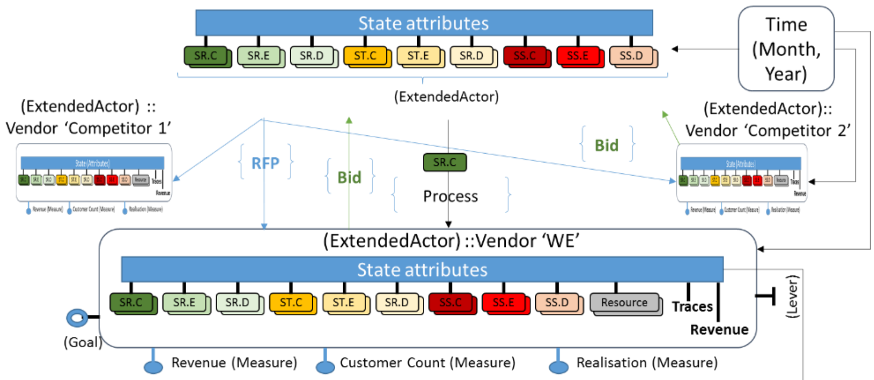
The next activity *Define GM-L* defines the Goal and Measures of identified OrgUnits, and map them with the Goals and Measures of problem entity that are identified in method step S1. In this example, the “WE” vendor owns the goals, measures and leavers defined in S1 method step. The generic Goals of Vendor and Customer are depicted in Figure 8 (b).

The remaining activity of the method step Conceptualisation of Organisation Model [S2] is *Define Behaviour*. This activity iterates over identified OrgUnits to define their behaviours. The typical Behaviours of Vendor and Customer are depicted in the form of state-machines in Figure 8 (b).

4.4 Implement Simulatable Model

Method step *Implement Simulatable Model* (manually) translates the information captured in method step S1 and method step S2 that collectively describe the Goal, Measure, Lever, Structure, Behaviour, State and Traces of OrgUnits into ESL specification by applying the transformation rules defined in Table 2.

A representative ESLMModel that contains two key ExtendedActors namely Customer and Vendor is shown in Figure 9. The Customer ExtendedActor comprises nine variables where each variables represents a bag of outsourced process of specific type from the business process classification *i.e.*, {SR, ST, SS} X {C, E, D}. The vendor ExtendedActor comprises Variables of Vendor OrgUnit that include State variables, Trace variables and the variables that represent Measures (as shown in Figure 9). The Customer and Vendor ExtendedActor also implement the state-machines depicted in Figure 8 (b).

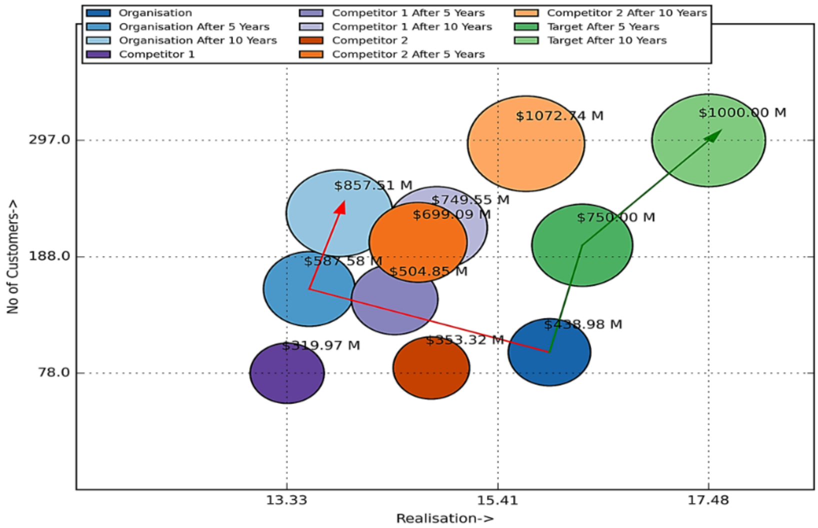


	Quardant	Billing Rate ($/Hr per FTE)		FTE Productivity	Negotiation Levers after term completion			Delivery Excellence
		Min	Max	Less FTE (in %) in Bid Quotation	Proposed FTE Reduction (%)	Proposal of Billing Rate Reduction (%)	Influencer Relation	Ration of [Excellent,Good, Normal, Below Normal]
Cost	Leader	8	12	8	2	5	Excellent	[60, 30,10, 0]
Efficiency	Visionary	18	22	6	0	5	Good	[50, 40,10, 0]
Delight	Contender	110	130	4	5	0	Normal	[20, 60,20, 0]

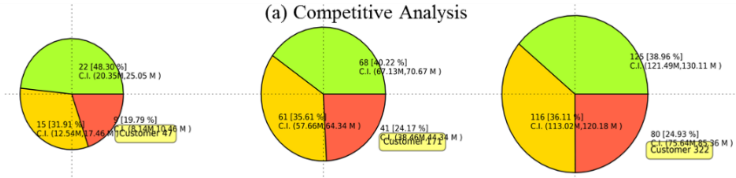
Figure 9 : Realisation of BPO scenario using EAMModel

The table in Figure 9 shows the initial characteristics of "WE" ExtendedActor. We make these Variables configurable to attenuate their values, thus these Variables also act as Lever specification in this example. As shown in the figure, a Vendor is equipped with a set of negotiation levers namely, the range of Billing Rate, range of FTE Productivity (percent reduction possible in number of full time employees), range of FTE Reduction (reduction possible during renewal of a contract), range of Billing Rate Reduction (reduction possible in billing rate during renewal of a contract), Influence Relation and Delivery Excellence. The Influence Relation is a qualitative characteristic that is quantified using four weighted labels namely 'Excellent', 'Good', 'Normal' and 'Not Good'. Value of Delivery Excellence attribute is a probability distribution. For instance, "WE" ExtendedActor is confident of delivering 'Excellent' quality on 60% of Cost kind of BPO projects won. The values for 'Good', 'Normal' and 'Below Normal' quality for this kind of BPO projects are 30%, 10% and 0% respectively. Therefore, one can model different kinds of vendors by setting appropriate values to the initial setting. The "Competitor" ExtendedActors are also modelled on the same lines as "WE" ExtendedActor.

The Customer ExtendedActor raises RFP events for outsourcing project. Each RFP event is characterized by the kind of process being outsourced (*i.e.*, SR or ST or SS), the objective for outsourcing (*i.e.*, C or E or D), size of the process in terms of FTE count, and the desired billing rate. Interested vendors respond to the RFP event by picking suitable values from their characteristics at random. Bid evaluation function is a weighted aggregate of the various elements of RFP response and a random value to capture effect of inherent uncertainty. The vendor with the best evaluated value wins the outsourcing process which gets executed as defined by the characteristics of the particular vendor. Essentially, an outsourcing process ExtendedActor moves from customer ExtendedActor to a vendor ExtendedActor (i.e., from customer basket to vendor



(a) Competitive Analysis



(b) Distribution of Sunrise, Steady and Sunset at “Now”, After 5 “Years” and after 10 “Years”

Figure 10 : Simulation Results when “WE” vendor continues as-is strategy

portfolio basket) as shown in Figure 9. The existence of an outsourcing process in a vendor portfolio impacts vendor’s State variable (and thus Measures) as outsourcing process contributes the Revenue, the customer count and Realisation. It also impacts the track record and market influences over the time.

The decision to renew existing contract is specified on similar lines but with a different set of characteristic variables influencing the decision. Essentially the autonomous outsourcing process ExtendedActor raises Renew event after 3 to 5 “Year” timeframe. Here too, the evaluation is cognizance of incomplete and uncertain knowledge renewability criteria.

4.5 Simulation

We use ESL simulator to simulate the business-as-usual operations of the “WE” vendor and its competitors. The simulation progresses with simulation ticks where each tick represents a “Month”. The outcome of simulation runs depicting possible states of “WE” vendor and its competitors at “Now”, after 5 “Years” and after 10 “Years” is shown in Figure 10 (a). As can be seen, the initial revenue of “WE” (represented using shades of ‘blue’ ellipses) is 438.98 MUSD from 90 customers with a realization of

Table 3. Decision Table

Lever	Revenue (MUSD)		Number of Customers		Realisation	
	After 5 Years	After 10 Years	After 5 Years	After 10 Years	After 5 Years	After 10 Years
No Lever	587.58	857.51	160	215	13.55	14
Improve Existing Resource	820.63	1165.80	195	287	15.2	15.4
Robotic Process Automation (RPA)	899.3	1309.87	201	301	15.3	15.7

nearly 15.5 USD per hour per FTE. Corresponding numbers for competitor 1 and competitor 2 respectively are < 319.97, 78, 13.33> (depicted using shades of 'violet' ellipses) and < 352.32, 79, 15.1 > (depicted using shades of brown ellipses). In short, at present "WE" vendor is doing much better than competition.

The graph, also shows the goals of "WE" vendor that aim to deliver <750, 200, 17> after 5 "Year" and <1000, 290, 18> after 10 "Year" (depicted using green ellipses). As can be seen, by continuing to operate the same way the "WE" vendor will be delivering <587.58, 160, 13.5> after 5 "Years" and <857.51, 215, 14> after 10 "Year" (as directed by red line in Figure 10 (a)) thus missing both the targets by a considerable margin. More importantly, competitor 2 will be overtaking "WE" vendor after 5 "Years" and both the competitors will be significantly ahead of "WE" vendor after 10 "Years".

Clearly, "WE" vendor cannot afford to continue with its current way of operation. A detailed analysis on portfolio of Sunrise, Steady and Sunset kinds of business processes, as shown in Figure 10 (b), indicates significant percentage of current revenue of "WE" vendor is from sunset kinds of outsourced processes (shown in red colour in Figure 10 (b)). Over time this market is going to shrink considerably as compare to the steady (depicted using yellow colour) as well as the sunrise (depicted using yellow green) business processes. Thus "WE" vendor needs to bring about a change in its characteristics so as to be able to win more bids in this demand situation.

4.6 Validation, Evaluation of Simulation Results and Recommendation

As part of model validation, we simulated the BPO specification by considering a known set of Vendors and Customers with fixed number of outsourced Processes. Es-

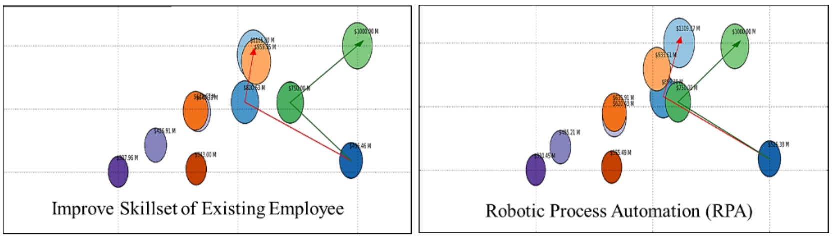


Figure 11 : Comparative study for Lever 1 and Lever 2

Table 4. Evaluation Summary

Requirement	EM Specification	Actor Lang.	Proposed Approach	Enabling Concepts in CM Model
Why	√	⊥	√	Goal
What	√	√	√	OrgUnit
How	√	√	√	Event and Behaviour
Who	√	⊥	√	OrgUnit
Where	√	⊥	⊥	OrgUnit
When	√	⊥	⊥	Time Event
Modular	√	√	√	OrgUni
Compositional	⊥	√	√	Composition Relationship
Reactive	⊥	√	√	IncomingEvent, OutgoingEvent
Autonomous	×	√	√	InternalEvent
Intentional	√	√	√	Goal
Adaptive	⊥	√	√	Adaptive Behaviour
Uncertainty	×	⊥	√	Stochastic Behaviour
Temporal	⊥	×	√	Temporal Behaviour
Measure Spec	⊥	⊥	√	Measure
Lever Spec	⊥	⊥	√	Lever
Top-down/ Bottom-up	Top-down	Bottom-up	Hybrid	Composition Relationship, Shared State Variable

Legends: √ : Supports adequately, ⊥ can be specified with difficulties, × : not supported

sentially we initialised Vendors and Customers to known states, simulated the specification for 2 "Years" and correlated observed simulation results with existing operational data to ascertain the validity of the constructed models.

After ensuring the operation validity of BPO specification, we explored two Levers as described in Figure 5 (b) and captured observed Measure values in the decision table as depicted in Table 3. Figure 11 and the decision table depicted in Table 3 show the comparative analysis of two Levers. With the Lever 1, the "WE" vendor is able to beat revenue target while failing to meet the number of customers and realization targets, whereas the 'WE' vendor is able to beat both revenue and number of customer targets while failing to meet the realization target narrowly with Lever 2. This clearly shows that the Lever 2 works well for "WE" vendor in the competitive environment described in this section.

5 Evaluation

For the kind of decision-making problem illustrated in this paper, industry practice relies extensively on spreadsheets, documents and diagrams. Such an approach typically represents the influence of Levers onto Measures in terms of static algebraic equations. However, value of a Lever and influence of a Lever onto a set of Measures can vary over time. This behaviour cannot be captured using spreadsheets. Neither there is any support for encoding stochastic behaviour.

The proposed approach enables modelling of a system of systems using a set of hierarchically composable OrgUnits each listening/responding/raising events of interest. Each individual system or OrgUnit encapsulates state (*i.e.,* a set of *State* variables), trace (*i.e.,* events it has responded to and raised till now) and behaviour (*i.e.,* encoding of individual reactions). They interact with each other by sending messages resulting into emergent behaviour (*i.e.,* the behaviour of system of system emerges from interactions of OrgUnits or systems). The proposed approach further helps in addressing the scalability issue by reducing the numerous message passing between OrgUnits through shared variables. Therefore, we claim the proposed approach provides primitives for creating models that closely mimic reality.

An evaluation of two prominent decision-making aids, *i.e.*, EM based approach and pure actor language based approach, along with presented approach is summarised in Table 4. As shown in the table, an EM based approach and an actor language based approach are complementary in nature. The former one supports aspect (*i.e., why, what, how, etc.*) specification and a top-down simulation approach, whereas actor language based approach is more effective for representing socio-technical characteristics and bottom-up simulation approach. But, it is not convenient for aspect specification. The proposed approach bridges the gaps between two classes of specifications by supporting comprehensive aspect specification and socio-technical characteristics as shown in Table 4. Moreover the explicit support for uncertainty, temporal behaviour, and the bottom-up and top-down combination make proposed approach suitable for CDDM.

6 Conclusion

Effective decision-making is a challenge that all modern organisations face. It requires deep understanding of aspects such as organisational goals, structure, operational processes. Large size, socio-technical characteristics, and increasing business dynamics make the decision-making a challenging task for the decision makers.

This paper argued that the efficacy of a complex dynamic decision-making (CDDM) chiefly depends on the three factors: (i) the availability of necessary and sufficient information in a machine-interpretable form, (ii) suitable machineries to process available information, and (iii) a method to capture information in a desired form and perform *what-if* analyses in a systematic manner. The paper presented an analysis of existing techniques and technologies to support a claim that the current state of the art decision making aids are inadequate for an affective CDDM and highlighted the gaps. Key aspects of this analysis point to the lacunae and inadequacy of support for representing necessary aspects of an organisation in a systematic manner, unavailability of appropriate concepts to represent the decision-making constructs, such as *Goal*, *Measure*, and *Lever*, and inability to handle inherent uncertainty. Importantly, the analysis also highlights the nonexistence of a suitable method supporting model construction, model validation and perform *what-if* analysis for effective CDDM.

To address these gaps, this paper contributed an approach that includes a meta-model to represent necessary and sufficient information in the form of a conceptual model (*i.e.*, CMModel), a meta-model to represent information in a simulatable form (*i.e.*, ESLMModel) and a method. The meta-model CMModel mitigates the identified specification gaps between the available technological capabilities and needs for CDDM (as highlighted in Table 1). The meta-model ESLMModel realises CMModel while addressing the analyses needs of CDDM. These models are supported and used by the proposed method that uses a top-down approach for defining goals, measure and levers (the GM-L structure), a middle-out approach for defining structural aspect of an organisation, and a bottom-up approach for behavioural specification, addresses methodical needs. The method, principally, combines a modelling and validation method defined by Robert Sargent [21] and a management sciences view for decision-making advocated by Richard Daft [3]. The method is evaluated through an industry scale case study from the BPO domain.

As part of future research, we intend to validate the proposed approach using real business scenarios as well as proposing further extensions to CMModel for introducing game theoretic approaches in simulations for CDDM. Other avenues of exploration include the use of constrained natural language to describe a problem entity so that a tool chain can be defined to automate production of the problem entity, conceptual model and the simulatable model. We expect the transformation chain to be human guided in the first instance.

References

1. Shapira, Z.: Organizational decision making. Cambridge University Press (2002)
2. McDermott, T., Rouse, W., Goodman, S., Loper, M.: Multi-level modeling of complex socio-technical systems. Procedia Computer Science 16, 1132{1141 (2013)
3. Daft, R.: Organization theory and design. Nelson Education (2012)
4. Conrath, D.W.: Organizational decision making behavior under varying conditions of uncertainty. Management Science 13(8), B-487 (1967)
5. O'Connor, T., and Hong Yu Wong. "Emergent properties." 2002.
6. Locke, E.: Handbook of principles of organizational behavior: Indispensable knowledge for evidence-based management. John Wiley & Sons (2011)
7. Iacob, M., Jonkers, D.H., Lankhorst, M., Proper, E., Quartel, D.D., Archimate 2.0 specification: Van Haren Publishing, 2012.
8. Peter Bernus ; Mertins, K. ; Schmidt, G., Handbook on architectures of information systems, ISBN 3-540-64453-9, 2006
9. Frank, U., "Multi-perspective enterprise modeling (memo) conceptual framework and modeling languages." HICSS. IEEE, 2002
10. Yu, E., Strohmaier, M., and Deng, X., Exploring intentional modeling and analysis for enterprise architecture. EDOCW, 2006.
11. OMG Document, Business Process Model and Notation, http://www.omg.org/spec/BPMN/2.0/, formal/2011-01-03, 2011.
12. Meadows, D.H., Thinking in systems: A primer. Chelsea Green Publishing, 2008.
13. Barat, S., Kulkarni, V., Clark, T., Barn, B.: Enterprise Modeling as a Decision-making Aid: A Systematic Mapping Study. PoEM 2016: 289-298.

14. Sandkuhl, K., Fill, H.G., Hoppenbrouwers, S., Krogstie, J., Leue, A., Matthes, F., Opdahl, A. L., Schwabe, G., Uludag, O., Winter, R.: Enterprise Modelling for the Masses - From Elitist Discipline to Common Practice. PoEM 2016: 225-240
15. Srinivasan, S., Mycroft, A., Kilim: Isolation-typed actors for java. In: European Conference on Object-Oriented Programming. 2008. pp. 104-128.
16. Haller, P., Odersky, M., Scala actors: Unifying thread-based and event-based programming. Theoretical Computer Science 410(2), 2009. 202-220
17. Allen, J., Effective akka. O'Reilly Media, Inc. 2013.
18. Agha, G.A., Actors: A model of concurrent computation in distributed
19. Zachman, J., et al., A framework for information systems architecture. IBM systems journal 26(3), 276-29, 1987.
20. Rumsfeld, D. Known and unknown: a memoir. Penguin, 2011.
21. Sargent, R.G., “Verification and validation of simulation models”. Winter simulation (pp. 130-143) 2005, December.
22. Barat, S., Kulkarni, V., Clark, T., Barn, B.: A Model based Realisation of Actor Model to Conceptualise an Aid for Complex Dynamic Decision-making. MODELSWARD 2017: 605-616
23. Thomas, M. and McGarry, F., “Top-down vs. bottom-up process improvement”. IEEE Software, 11(4), 1994. pp.12-13.
24. Beckermann, A., Flohr, H. and Kim, J. eds., “Emergence or reduction?: Essays on the prospects of nonreductive physicalism”. Walter de Gruyter. 1992.
25. Camus, B., Bourjot, C., Chevrier, V., Combining devs with multi-agent concepts to design and simulate multi-models of complex systems. In: Proceedings of the Symposium on Theory of Modeling & Simulation 2015, pp. 85-90
26. Siebert, J., Ciarletta, L., Chevrier, V.: Agents and artefacts for multiple models co-evolution: building complex system simulation as a set of interacting models. In: 9th International Conference on Autonomous Agents and Multiagent Systems: 509-516. (2010)
27. Borshchev, A., The big book of simulation modeling: multimethod modeling with AnyLogic 6. North America Chicago
28. Macal, C.M. and North, M.J., 2010. “Tutorial on agent-based modelling and simulation”. Journal of simulation, 4(3), pp.151-162.
29. Rolland, C., Selmin N., and Georges G., Enterprise knowledge development: the process view. Information & management 36.3 (1999): 165-184.
30. Sandkuhl, K., et al., Enterprise Modeling. Tackling Business Challenges with the 4EM Method. Springer 309 (2014).
31. van Langevelde, I., Philipsen, A. and Treur, J., Formal specification of compositional architectures. 10th European conference on Artificial intelligence, 1992.
32. Bock, A., Frank, U., Bergmann, A., Strecker, S., Towards Support for Strategic Decision Processes Using Enterprise Models: A Critical Reconstruction of Strategy Analysis Tools. PoEM 2016: 41-56
33. David, K., Georgeff, M., and Rao, A.: A methodology and modelling technique for systems of BDI agents. Agents breaking away,1996.
34. van Langevelde, I., Philipsen, A. and Treur, J., "Formal specification of compositional architectures." 10th European conference on Artificial intelligence, 1992.
35. Bock, A., Frank, U., Bergmann, A., Strecker, S., Towards Support for Strategic Decision Processes Using Enterprise Models: A Critical Reconstruction of Strategy Analysis Tools. PoEM 2016: 41-56
36. David, K., Georgeff, M., and Rao, A.,: A methodology and modelling technique for systems of BDI agents. Agents breaking away,1996.

37. Hewitt, C.: Actor model of computation: scalable robust information systems. arXiv:1008.1459.
38. Langley, A., et al. "Opening up decision making: The view from the black stool." organization Science 6.3 (1995): 260-279.
39. Kulkarni, V., Barat, S., Clark, T., Barn, B.: Toward overcoming accidental complexity in organisational decision-making. In: Model Driven Engineering Languages and Systems (MODELS). pp. 368-377 (2015)
40. Barat, S., Kulkarni, V., Clark, T., Barn, B.: A simulation-based aid for organisational decision-making. In: ICSOFT-EA 2016: 11th International Conference on Software Engineering and Applications (2016)
41. Kulkarni, V., Barat, S., Clark, T., Barn, B.: Using simulation to address intrinsic complexity in multi-modelling of enterprises for decision making. In: Proceedings of the Conference on Summer Computer Simulation. pp. 1-11 (2015)