Early Experience with System Dynamics Modeling for Organizational Decision Making

Asha Rajbhoj
Tata Consultancy Services Research
Pune, India
asha.rajbhoj@tcs.com

Krati Saxena
Tata Consultancy Services Research
Pune, India
krati.saxena@tcs.com

ABSTRACT
Increased business dynamics mandates modern organizations to proactively prepare their response to operating environment changes. Typically, large size organizations consist of multiple interconnected departments. Individual department strategies need to be balanced so as overall organization performance is improved. With increased number of interdependent strategic parameters, complexity of their combinatorial evaluation increases. System dynamics (SD) modeling supports modeling dynamic behavior of organizations and strategy validation through simulation. We used SD modeling techniques to model large size organization decision making problem that involves dynamic as well as combinatorial complexity. In this paper, we share our experience and learning from this endeavor.

CCS Concepts
• Organizational decision making→System dynamic modeling→ combinatorial complexity • Enterprise modeling →Simulation.

Keywords
Organizational decision making; Enterprise modeling; System dynamics modeling; Simulation

1. INTRODUCTION
Today's business environment is characterized by its dynamic nature. To survive and remain competitive, modern organizations need to sense the environment changes and respond to them proactively [1]. Typically large organizations have multiple inter-dependent departments each focusing on specific organization function. Each department has different, multiple strategies to optimize performance. Strategy adopted by individual department keeping local optimization focus does not lead to overall organization optimal performance. Individual department strategies need to be balanced so as overall best organization performance is achieved [2]. This is possible only through con-joint evaluation of all related strategic parameters. As number of such parameters increases their combinatorial evaluation complexity also increases. Thus, considering all inter-dependencies and finding best possible response to changed operating condition is an effort-, time- and cost-intensive, and error prone activity. Also, cost of erroneous response is prohibitively high. Hence, it is necessary to evaluate strategies before their adoption. Simulation based approaches supporting temporal and quantitative analysis are best suited for such needs.

System dynamics (SD) modeling [3, 4] supports modeling dynamic behavior of organizations and early validation of strategies through temporal and quantitative analysis. Main concepts in SD modeling approach are causality, feedback loop, stock (accumulation of entities), flow (dispersal of entities) and delays. Many SD tools [5, 6] are available that provide sophisticated simulation support to play out various what-if scenarios. SD modeling approach has also been proposed in decision making of many domains [7, 8, 9]. These uses mainly highlight dynamic complexity in decision making. We used SD modeling approach for large scale organizational decision making involving both dynamic and combinatorial complexity. We modeled fairly large IT services provisioning organization from practitioner perspective and simulated models for 5 years to evaluate various strategies related to revenue and profit growth under different operating environment conditions and changing organization state. We share our early experience and learning from this exercise in this paper. Though overall experience is shared in a specific context, we believe, enterprise architects, researchers and practitioner will find the takeaways from this experience applicable even in a more general context.

The rest of the paper is organized as follows. Section 2 of the paper presents a motivating example. Section 3 describes models and analysis results of the example. Finally, we discuss our experience, learning and future work in section 4.

2. MOTIVATING EXAMPLE
In this section we introduce a motivating example that sets the context for the rest of the paper. Let us consider a large IT services provisioning organization whose main business is developing software projects as per customer needs. Overall high level activities involved in this business are ‘bid project’ -> ‘win project’ -> ‘execute project and maintain good track record’ -> 'receive payment from customers’. For executing these activities there are multiple departments involved as shown in Fig 1. Sales department bids for projects by responding to requests for proposals (RFPs). Resource management department recruits people and allocates people for project execution. Delivery department executes projects. Account department keeps track of finances. Each of these departments may choose different strategies for achieving their goals as shown in Fig.1. For instance to win more projects sales department may offer to reduce bid price, promise early delivery etc. Delivery department may consider improving employee productivity to maximize timely

delivery. Resource management function may offer better pay package to improve joining probability and so on.

These departments are interdependent on each other as shown in Fig.1 and Fig. 2. Good delivery track record helps sales department win bids. Delivery department cannot function unless sales win projects and resource management department makes appropriate number of people available. Unavailability of people leads to delayed start of project that in turn results in delayed delivery and penalty. Also, people cannot be recruited in excess as more people on bench leads to poor employee utilization thus impacting profitability of the organization. Functioning of these departments is also influenced by external events on which organization does not have any control e.g. RFP arrival rate, supply of people etc – moreover they may vary over time. Factors such as RFP win rate, employee attrition etc are also influenced by size and quality of competition. Organization state also keeps on changing over time. For instance, employee's experience in executing projects, senior-junior ratio, salary costs, organization delivery track record etc. is dynamic. Thus, overall dynamics in operating environment as well as changing organization state complicates decision making process. Given this organization dynamic context, organization has to analyze its performance and various actions in response to changes in different operating environment conditions. As an example in this paper we considered changing demand situation and evaluated various strategies to analyze organization performance in terms of revenue and profit growth over 5 years.

IT services provisioning organization also has to face continuously rising demand for reduced price and/or reduced time-to-market delivery. To stay relevant in business, organization has to continuously improve its operating efficiency. Hence, organization is looking forward to develop end-to-end code generation tools and use them for project development. Before implementing this strategic decision, it would like to first analyze: what will be the overall investment to develop end-to-end code generator tools? , will this approach be profitable? how long will it take to reach break-even point for profitability?.

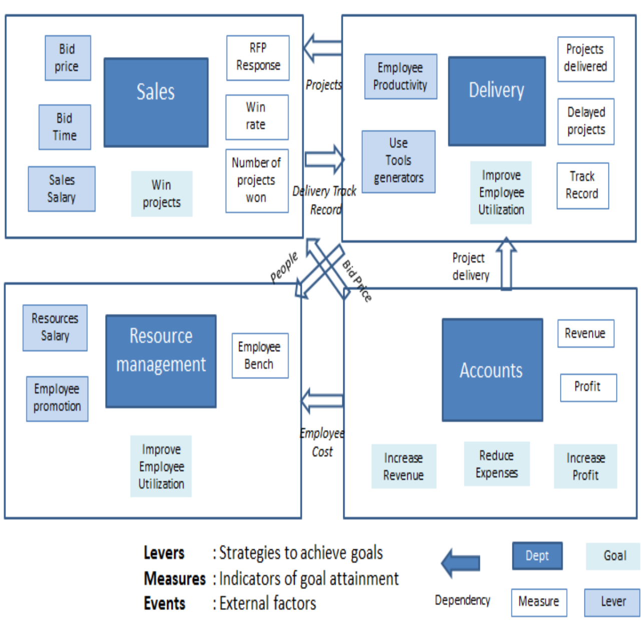

Fig 1: Organization Structure

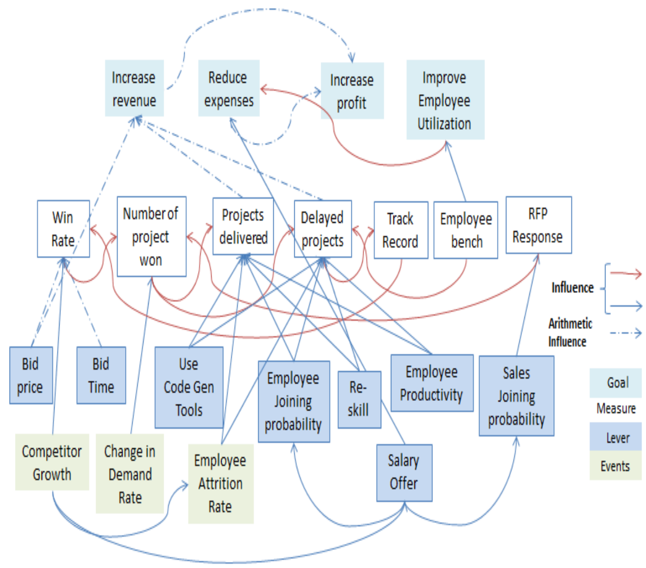

Fig 2 : Goals, measures, levers influence relationship

3. MODEL BASED DECISION MAKING

We used system dynamic modeling iThink [5] tool to model organization behavior and analyze questions of interest. For creating models we have made certain assumptions about organization settings. Typically, service provisioning organization executes projects of different kinds, sizes, and complexities. We considered 2 kinds of projects namely J2EE and Mobile. Each of these project kinds are further classified on size (**S**mall/**L**arge) and complexity (**S**imple/**C**omplex) dimensions. Thus there are 4 types i.e small simple (SS), large simple (LS), small complex (SC), large complex (LC) of J2ee projects and similar 4 types of mobile projects considered. We used COCOMO [10] equations for estimating the effort, time, team size. For each type of the projects, arrival rate, winning rate, pricing etc. are given as input. Typically, project execution resources have different kind of skills and different number of years of experience. We considered two kinds of workforce resources i.e. Junior (J) and Expert (E) having different productivity and salary. Initial state of the organization is specified by setting the initial values of all the lever variables, other internal variables and stocks as shown in Table 1.

Table 1 : Variable values

Variable	Value	Variable	Value
Employee count	1600	Trainee salary (USD PM)	1K
Junior distribution	70 %	Junior salary (USD PM	1.1K
Expert distribution	30 %	Expert salary (USD PM	3 K
Initial sales count	150	Trainee joining probability	70%
SS price / arrive rate	200 / 10	Expert joining probability	50%
LS price / arrive rate	600 / 8	Sales joining probability	50%
SC price / arrive rate	330 / 8	Junior attrition rate	10 %
LC price / arrive rate	1600 / 6	Expert attrition rate	10%

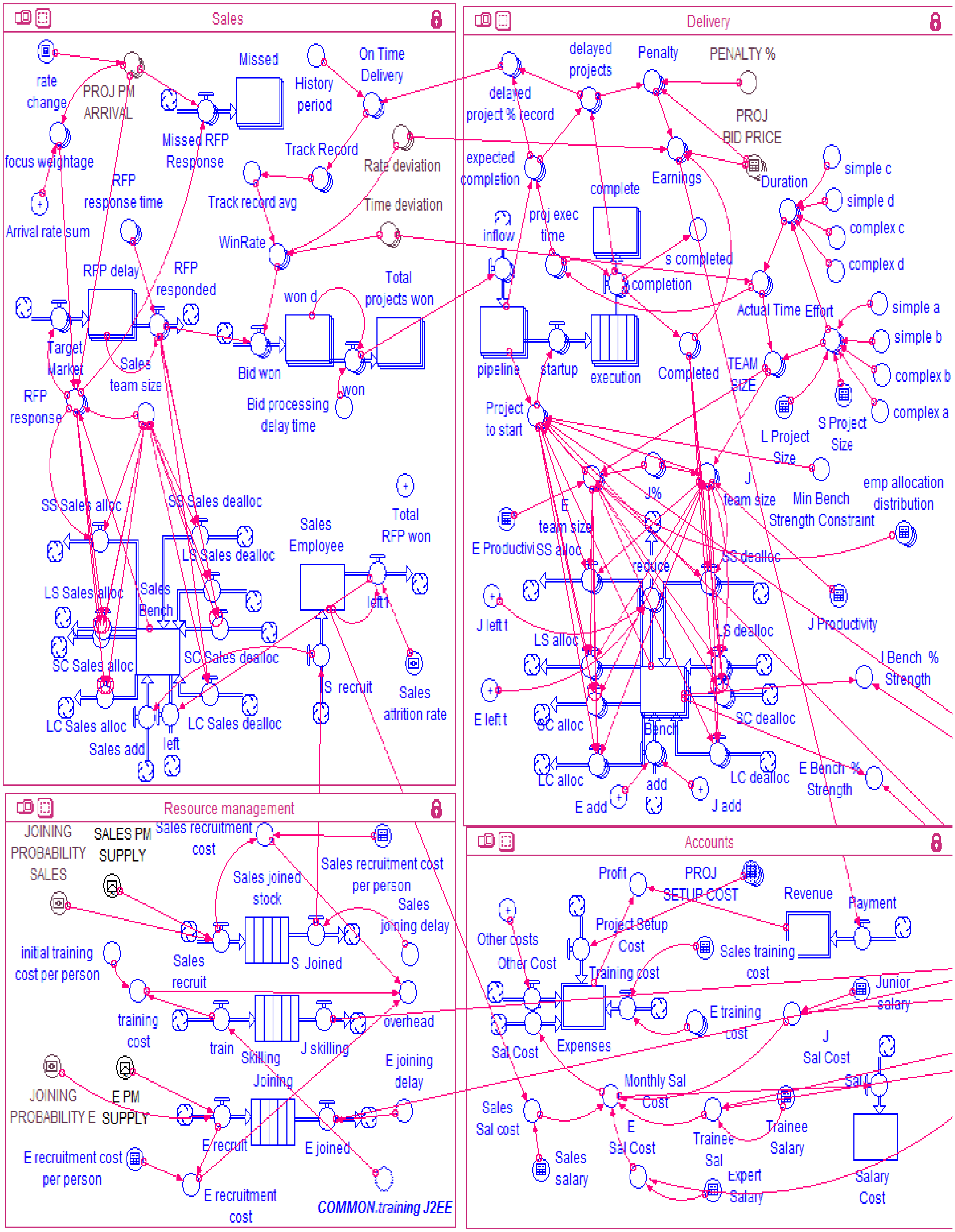

Fig. 3: System Dynamic Model

3.1 Analyzing organization growth under dynamic operating conditions

In the interest of space, we discuss only a subset of the model. Fig. 3 shows systems dynamic model describing behavior of sales, delivery, resource management and accounts department. '*Sales'* sector covers project bidding process where '*PM arrival rate'* and 'S*ales team size'* variables control the project '*Target market*' inflow. RFP response time is modeled using *'RFP delay'* stock and using delay function on outflow *'RFP responded'*. Similarly bid processing delay is modeled using a separate stock and delay function. The stock *'Missed'* represents missed RFP opportunities. This measure helps in deciding the sales team size. Bid win rate is modeled as *'WinRate'* variable which is computed using track record, rate deviation, price deviation variables. Delivery sector covers project execution process wherein '*pipeline'* stock represents bids won. '*Project to start'* variable determines the flow of project that can start execution. Project people allocation is decided using this variable. For each J and E bench separate stocks are used and project allocation is modeled using outflow and de-allocation is modeled using inflow. Allocation, de-allocation part covers people allocation to projects considering project size, projects in pipeline, bench strength and project execution priorities. People are de-allocated and moved to bench on project completion. On promotions J people are moved from J stock to E stock. '*Delivery'* sector also shows delayed projects. We only considered the delay due to unavailability of people. To model different types of projects, we used array abstraction [11] in sales and delivery sector. 'Resource management' sector shows recruitment of different type of people and delays. Joining delay of people for all type resources is modeled using conveyor stock. *Accounts* sector shows profit, revenue, expenses computations. Salary, training, project related expense are accumulated in '*Expenses'* stock. Earnings from projects are accumulated in '*Revenue'* stock. Profit is computed using values of these stocks. Competition influence to the organization impacts win rate, employee attrition rate, employee joining probability parameters. It is modeled by setting appropriate initial value of these parameters

3.1.1 Results

Once all the necessary behavioral aspects are captured we played out different what-if scenarios. Prior to analysis of dynamic demand, we analyzed whether organization is operating in comfort zone and is there any scope for improving employee utilization. With given initial settings, 5-year simulation showed significant value for *'J bench'* and *'E bench'* stocks. Thus, it can be inferred that organization can target more number of projects say by increasing bid win rate. For achieving this possible strategies are: To reduce time by 10%, to reduce price by 10% or reduce time and price both. We observed *reduce time* gave better revenue and profit growth as compared with reduce price. Use of both strategies together increased the winning rate even more. Undesirable effect was significant increase in number of projects witnessing delayed start due to unavailability of resources – value of *J bench* and *E bench* stocks. The delayed start resulted in delayed delivery and subsequent penalties. To reduce delays we used strategy to improve employee productivity through training. With this lever change, project delay got reduced and revenue and profit increased. We observed bench measures again to check further possibility of improvement. Bench of J people indicated that its going quite low as compare to E people. To keep balance of senior:junior ratio, possible solution was to increase J people salary to increase the joining probability. Hence, we used strategy to increase pay for junior to 1.2K. With this change, delayed projects are reduced to zero , but increase in revenue was marginal and profit was reduced to great extent. Thus, this analysis hinted that strategy to reduce time and to improve employee productivity through training augurs well for growth.

Using reduced time and improved productivity option, we analyzed next scenario of changing demand situation. We considered slow decline of traditional J2EE project demand and steep rise in of mobile project demand as shown in Fig. 4. X axis shows time duration of 5 years. Y axis shows arrival rate change with maximum value being 11 projects per month. The project arrival rate change is considered non-linear and different for different types of project as shown in the Fig.4. With demand change, we observed that number of delayed mobile projects have increased and mobile J, E bench gone to low peak. As J2EE project demand reduced, it's J, E bench was increased. Hence, J2EE people can be reskilled to Mobile. To arrive at appropriate reskill rate, we played out with different value as shown in Table 2.

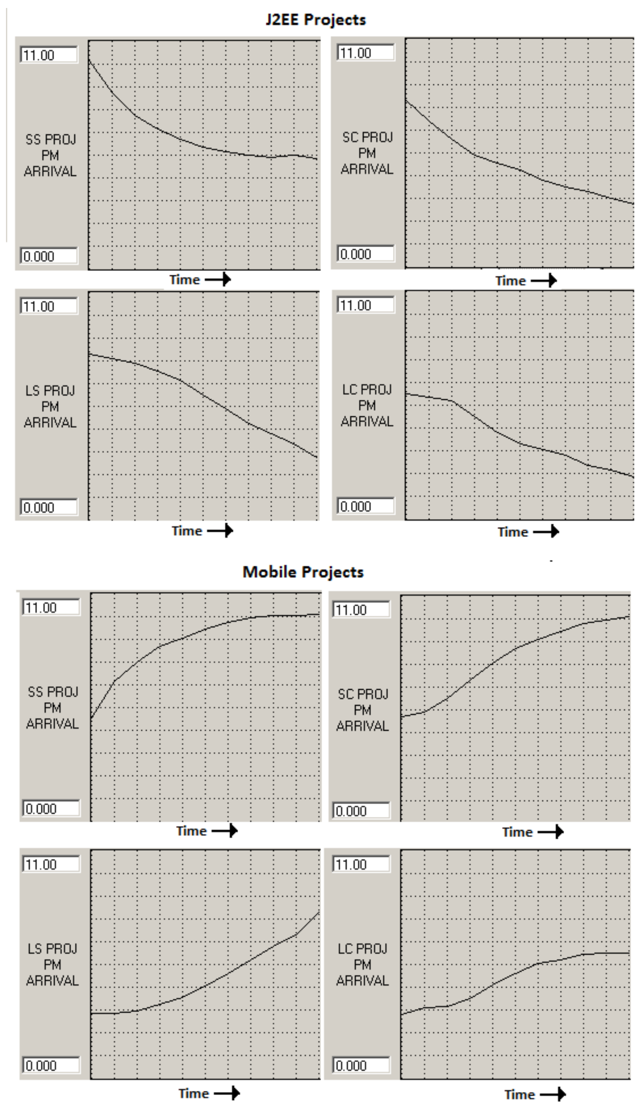

Fig 4: Demand rate change

Table 2 : Re-skill rate change scenario playing

E reskill rate	J reskill rate	Trainee %	Revenue	Profit	
0	**0**	**75**	**359251**	**152983**	Initial values
20	12	70	266194	61521	Degraded results
15	8	65	349402	144532	
13	6	65	363441	158366	
11	5	65	368230	163036	
7	3	65	371264	165831	
11	5	70	372075	166652	
10	4	70	372817	167292	
9	**4**	**70**	**373048**	**167505**	Better results
2	**3**	**70**	**373259**	**167509**	

For the given organization state, E reskill rate of 2 per month and J reskill of 3 per month gave better result. Delayed mobile project were also reduced. For further possibility of improvement when we observed mobile '*Missed*' opportunities measure, it indicated that there is still more market that can be targeted to utilize J2EE J,E bench strength. Hence, we increased mobile sales people by increasing their joining probability through increased payment. We played out various reskill rates again as earlier. E reskill rate of 13 per month and J reskill of 6 per month and allocating 65 % trainee to Mobile projects gave better revenue and profit results. Thus, through simulation we could arrive at reskill rate for given changing demand rate scenario.

3.2 Analyzing code generator platform development investment

For analyzing whether investing in developing end to end code generator [12] tool be profitable and by when break-even point of profitability will be reached we extended the system dynamic model shown in Fig.3. We added code generation based project execution in delivery model. Generation based project development time is computed using COCOMO estimation. We further extended model to capture tool development effort and delays as shown in Fig. 5. '*Tool use %*' indicated percentage of total won projects that can be targeted using generative approach. Use of generative approach also introduced dynamics in organization operation. Use of generative approach also influenced targeted projects. Due to productivity gain, number of people required for project execution were reduced; as a result more people were available for project execution; hence more projects could be targeted which in turn increased tool usage. This created positive re-enforcing cycle for organization operation.

We assumed that there is no direct revenue earning through generator tools sell and gains are primarily due to indirect profit earning due to productivity gain and competitive advantage for project win. To analyze expenditures, salary of the all employee allocated for the tool development is accumulated in '*Tool dev expenses*' stock. To analyze gains, employee effort saving for execution of projects using code generation approach is transformed to indirect money saved and accumulated in '*Tools indirect revenue*' stock. '*Project completed using tools*' is dynamic entity in delivery model. It is used for computing indirect savings per project.

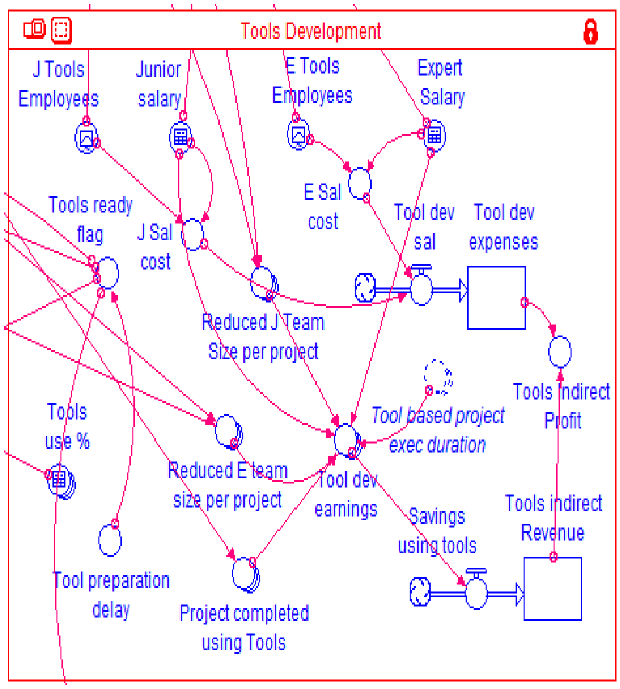

Fig. 5 : Break even analysis model

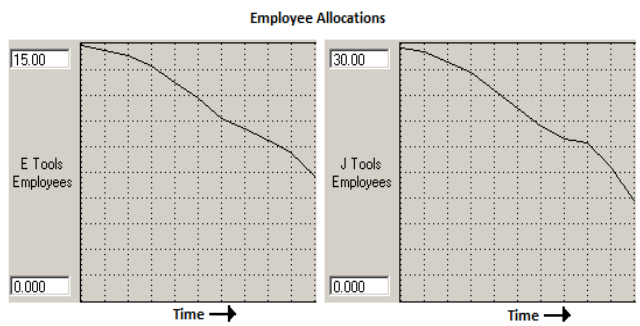

Fig. 6: Employee allocations for tool dev

3.2.1 Results

For developing end to end code generation tools we considered initial 30 J employees and 15 E employees with reducing strength over 5 year as shown in Fig. 6. X axis shows time duration of 5 years. Y axis shows employee allocations for the tool development. We arrived at this tool team allocations considering following assumptions: 1) Initial tool development time as 6 months. 2) Post development small % of team participates in initial tool deployments and large % of team continues with tool enhancements for betterment. 3) Over 3 years there will be continuous increase in tool use and tool enhancement effort will be slowly reduced. 4) Towards end major effort goes in tools deployments, support and consultancy. With these assumptions overall team size is considered reducing over 5 years.

With these settings, when we ran the model for 5 years we observered that break-even point is reached after 45 months as shown in Fig.7 and thereafter profit gain was exponential. Thus, developing end-to-end code generation tools and using them for project development validated as a viable option for the given settings.

Fig. 7 : Break-even analysis result

4. EXPERIENCE, LEARNING AND FUTURE WORK

Using system dynamic modeling we could model behavior of various departments of large IT services provisioning organization. Quantitative nature of SD models helped in specifying operating environment conditions in terms of RFP arrival rate, supply of people, and competition influence and so on. We could also model dynamic organization state in terms of people bench, different skilled people, expertise level (J / E), and organization track record and so on. SD models are typically meant for aggregated and generalized view of systems. With people and project type use, we tried to make model more specific and bit closer to real life situation. These types can be expanded further for improving accuracy of the results; however as it hampered simulation speed and also increased further combinatorial complexity we restricted it to few types.

Prior to SD modeling we identified various levers, measures, goals and their interrelationship. This activity was solely manual and required necessary expertise about the domain of the organization to be modelled. This representation helped in the problem space definition clarity and in creating SD model. It also guided about which measures to be observed and next strategic value / option to be chosen during simulation.

Evaluation of organization goal attainment under different operating conditions was possible by playing out different strategic parameters con-jointly. However, we observered that playing out multiple levers to arrive at suitable strategic option was very time consuming activity. Our observation is in line with Homer [13] that says "The more one extends the scope of a model in an attempt to make it more useful or complete; expanding it to include more concepts and variables of interest, the more effort will be required to achieve a desired level of evidence". Each simulation run involved changing lever values -> observing measures -> getting hints about what should be value of same lever and / or next lever change . This loop had to be repeated multiple times till best possible lever values are arrived at. Numbers of change levers were more, and most of the levers were of integer value types which increased the simulation search space to large extent and resulted in increasing combinatorial complexity. Moreover, changing lever values and observing simulation result was completely manual process hence it look considerable amount of time and effort. We would say our example covered comparatively few set of levers. Typically, large organizations play with many more number of strategic levers to analyze operating environment dynamics. Hence, we think both dynamic and combinatorial complexities are important concerns for organizational decision making. SD modeling only addressed dynamic complexity in decision making of IT services provisioning organization. Our initial study indicates, hybrid approaches that can combine other technique with SD modeling technique [14, 15] may help in addressing combinatorial complexity also. We planned to explore further in this direction.

5. REFERENCES

[1] Soroosh Nalchigar, Eric S. K. Yu, Steve M. Easterbrook. Towards Actionable Business Intelligence: Can System Dynamics Help? PoEM 2014: 246-260

[2] Suman Roychoudhury, Asha Rajbhoj, Vinay Kulkarni, Deepali Kholkar. Models to Aid Decision Making in Enterprises. ICEIS (3)) 2014: 465-471

[3] Jay Wright Forrester. Industrial dynamics, volume 2. MIT press Cambridge, MA, 1961.

[4] John Sterman. Business dynamics. Irwin-McGraw-Hill, 2000.

[5] http://www.iseesystems.com/Softwares/Business/ithinkSoftware.aspx

[6] Vensim, November 2013. http://www.vensim.com.

[7] Francisco Campuzano and Josefa Mula. Supply Chain Simulation: A System Dynamics Approach for Improving Performance. London: Springer, 2011, vii, 106 pages. ISBN 978-0-85729-718-1

[8] Hirsch G, Homer J. Modeling the dynamics of health care services for improved chronic illness management. In: 22nd International System Dynamics Conference. Oxford, England: System Dynamics Society; 2004.

[9] Pruyt, E. (2006a). System dynamics and decision-making in the context of dynamically complex multi-dimensional societal issues. In Proceedings of the 2006 Conference of the SystemDynamics Society, Nijmegen. System Dynamics Society. 7, 12

[10] Barry W Boehm. Software Engineering Economics. Prentice Hall, 1981 ISBN:0138221227

[11] Myrtveit, Mangne. Powerful Modeling Using Array Variables. 12th International Conference of the System Dynamics Society

[12] Vinay Kulkarni, R. Venkatesh, Sreedhar Reddy: Generating Enterprise Applications from Models. OOIS Workshops 2002: 270-279.

[13] Homer J. (2014), Levels of evidence in system dynamics modeling, Syst. Dyn. Rev., 30: 75–80. DOI: 10.1002/sdr.1514

[14] Duggan, Jim. 2007. Using system dynamics and multi objective optimization to support policy analysis for complex systems. In Understanding Complex Systems, 59–81. Springer Berlin/Heidelberg.

[15] Alborzi, M., 2008. Augmenting system dynamics with genetic algorithm and TOPSIS multivariate ranking module for multi-criteria optimization, Proc. of Int. Conf. of the System Dynamics Society.