Governance Policies in IT Service Support

Abhinay Puvvala and Veerendra K. Rai
Tata Consultancy Services, 54B, Hadapsar Industrial Estate, Pune- 411013, India

Keywords: IT Service Support, Agent based Model, Governance Policies, Upper Confidence Bound Algorithm.

Abstract: IT Service support provider, whether outsourced or kept in-house, has to abide by the Service Level Agreements (SLA) that are derived from the business needs. Critical for IT Service support provider are the human resources that are expected to resolve tickets. It is essential that the policies, which govern the tickets' movement amongst these resources, follow the business objectives such as service availability and cost reduction. In this study, we propose an agent based model that represents an IT Service Support system. A vital component in the model is the agent 'Governor', which makes policy decisions by reacting to changes in the environment. The paper also studies the impact of various behavioural attributes of the Governor on the service objectives.

1 INTRODUCTION

Communicating high level business objectives and their relative importance from the IT Governance to the IT Operations is a challenging task. The measure of this challenge lies in understanding that Business-IT alignment or the lack of it still remains one of the major corporate concerns and the most critical measure of the success of information technology as a value adding component of business enterprises. It is imperative that the design of each IT system is aligned to business objectives without compromising on the efficiency of IT systems (Sallé, 2004). In this study, we focus our attention on the IT Service support system. Specifically, this paper proposes a framework for the governance of IT Service support engagements.

Governance is a mechanism of course correction when a project, program or engagement is in execution mode to help projects meet the intended outcomes. Scope of governance includes, among other things, structural and organizational changes, communications and policies. The scope of governance in this study is limited to a set of rules (policies) that includes assignment rules and pre-emption rules in IT Service support engagement. These rules play a key role in realizing the intended objectives of the engagements.

Owing to the volatile nature of IT service support engagements environments, these rules cannot be set a priori and be expected to remain optimal throughout the course of an engagement. Given the constantly evolving business needs and their possible repercussions on the IT systems, it is not feasible to have a static set of rules. Another key consideration while determining an optimal set of rules is the interdependencies between them. We propose an agent based game theoretic approach to derive the optimal set of rules (assignment, escalation and pre-emption rules) based on the objectives and the context of a support engagement.

The rest of this paper is organized as follows: Section 2 contains survey of relevant literature, research gaps and the contribution of this study. Section 3 describes the research model along with discussions on relevant concepts from literature. Section 4 contains the illustration of model proposed in Section 3. Section 4 also has results of what-if and sensitivity analyses. Section 5 concludes the paper.

2 LITERATURE REVIEW

Studies related to the governance of IT Service support have primarily focussed on the following research questions: 1. how to prioritize tickets based on the business needs, thereby indirectly focusing on the assignment rule, and 2. how to optimally divide the staff amongst multiple shifts and technology towers. While some studies (Gurvich et al, 2007; Bassamboo et al, 2004) have attempted to address these questions together, in most of research studies

Puvvala, A. and Rai, V.
Governance Policies in IT Service Support.
DOI: 10.5220/0006312805120519
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1*, pages 512-519
ISBN: 978-989-758-247-9
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

these questions have been addressed independently. Of these two questions, the focus of this study is closer to the former rather than the latter. Bartolini and Salle (2004) have proposed an approach to present how business needs are used to prioritize tickets and allocated to human resources.

In practice, basic and intuitive assignment policies such as First Come First Serve (FCFS) and priority based FCFS are often put to use. While FCFS follows a strict first come first serve routine, priority based FCFS gives precedence to requests with higher priority. While these policies are intuitive and easy to implement, they do not consider the SLA norms, penalty costs etc. Assuncao *et al* (2012) have studied the impact of both assignment and pre-emption policies on ticket resolution and service level agreement attainment. The dependence of the policy optimum on the distribution of workload is evident in their study. Lunardi *et al* (2010) also have studied the management of changes in the domain IT service support. Beyond that, there is a vast amount of literature in the domain of operations research on task scheduling (Rothkopf 1966; Pinedo 1995) in manufacturing that can be drawn upon.

As mentioned in section 1, we use Agent Based Modelling (ABM) to represent the engagement. Agent-based modelling is an effective simulation modelling technique that has grown rapidly in the last few years. Agent based modelling is considered a powerful paradigm to model human centric systems like IT service support (Bonabeau, 2002). The basic tenet of ABM is that a collection of autonomous decision making agents that produce emergent behaviour by interacting in an environment under a given set of rules (Davidsson, 2002). This view resonates well with IT service support system where various agents such as tickets and resources who individually interact with each other under defined processes which in turn are a result of the policies. These interactions are analysed by simulation of the agents' behaviour. It is a relatively new and emerging method in social sciences, which can be applied to a problem by defining a set of agents with related attributes, behaviours and fitness function, the simulation environment and the overall performance-measuring objectives of the environment (Mataric, 1993).

A typical ABM model consists of an agent having certain attributes, rules/actions, goals and decisions to make. These defined agents are generally governed by a fitness function. The aim of creating a fitness function is that it allows multiple agents of similar nature to have different attributes by creating differences in parameters of fitness function. This heterogeneity thus created is an essential component of ABM and helps mimic the real world more closely than other methods. These countless interactions lead to 'emergence' of new behaviour which had not been programmed into the behaviour of the individual agents (Waldrop, 1992). Agent based modelling has already been extensively used in economics (Agent Based Computational Economics (ACE)). Zaffar *et al* (2008) used it to identify the impact of Variability of Open Source Software (OSS) support costs, length of upgrade cycle and interoperability costs on OSS diffusion. Applications of Agent based modelling in IT systems are limited. Jha *et al* (2014) have proposed an agent based approach for estimating effort required to resolve incidents in an IT support engagement. In the next section, we discuss how agent based modelling has been used to model IT service support engagement.

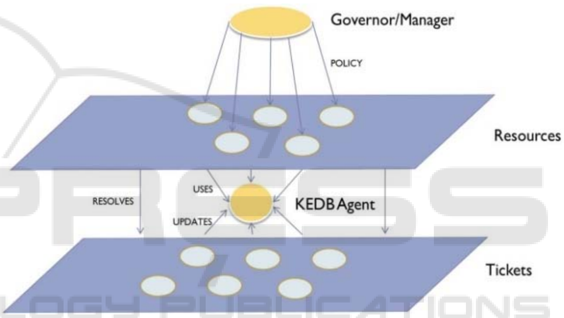

Figure 1: Model Topology.

3 MODEL

3.1 Agents

Although the literature on Agent based modelling does not provide a clear cut consensus on the approach to identify agents in a system, there are some basic guidelines that are common across various definitions of agents. Bonabeau (2001) considers any entity that has independent behaviour governed by very basic reactive decision rules to a complex and adaptive artificial intelligence. In contrast, some researchers emphasize on the ability of these entities to be adaptive to the environments and have a learning component ingrained in their behaviours. Casti (1997) separates these behavioural elements into base level and higher level rules. While the base level rules are meant to respond to the environment, higher level rules can dynamically change the base level rules (rules to change the rules) by learning and adapting to the environment. Jennings (2000) too emphasizes the need for presence of agent attributes that are active rather than purely passive. Active agent attributes are essential for autonomous behaviour by which agent can make independent decisions.

We adopt Bonebeau's (2001) view to identify agents in this system. The agent-set is a mixture of autonomous, semi-autonomous and dependent agents. Figure 1 has the topology of the agent based model used in this study. Each agent is explained in detail in the following sub sections.

3.1.1 Tickets

In IT production support, a ticket is an abstract unit of work. Ticket can be any one of events, incidents, problems, access requests and change requests. Based on the business needs, tickets have to be handled within specified time as directed by the Service Level Agreements (SLAs). Typically, the SLA terms are dependent on the ticket's priority. Priority is a composite of the urgency of the ticket (how soon the business needs to be resolved) and impact (how many users are affected by the ticket). Also, tickets vary based on the type of skill required for resolution. A "Technology Tower" signifies a method of work organization usually employed in IT production support where issues are grouped along technical domains. Examples of technology tower could be ".net", "Java", "SQL" etc. Naturally, the skills needed to resolve tickets belonging to each of these technology towers are different. While the type of skill needed for resolution determines the ticket's technology tower, difference in level of skill needed for resolution determines the level of support tickets are routed to or eventually escalated to.

Since ticket handling is a knowledge intensive task, a repository of all the information known about tickets is maintained. The repository can take the form of a set of standard operating procedures (SOPs) or entries in Known Error Database (KEDB). The effort needed to resolve tickets has been observed to follow Power Law Distribution (PLD). Based on the above described characteristics the list of ticket attributes are shown in Table 1.

3.1.2 Resources

Despite the ongoing drive towards automation, ITSM is majorly a human centric system. Tickets are handled by resources, which are categorized into multiple teams based on their skills and specializations. In a typical IT production support setup, tickets are responded and resolved by resources. Response includes identifying, logging, categorizing, prioritizing, routing and conducting initial diagnosis of tickets. Whereas, Resolution relatively is a more complex task. It involves performing a set of steps to resolve a ticket. And, it is done at the level of support that corresponds to the ticket's required resolution skills.

As given in Table 1 resources are characterized by a set of static and dynamics attributes in our model. While, technology tower, competency, cost, likelihood of absence are the attributes that remain static over the simulation. In contrast, ticket, shift, net effort are the attributes that change dynamically with the environment. The support structure support in the model comprises of two technology towers with teams divided into three and two levels of support. Further, the support service is to be provided 24x7, divided into 3 shifts of 8 hours each..

3.1.3 KEDB Agent

It is critical for any IT production support engagement to record the knowledge acquired by human resources in the process of handling tickets to the extent possible. Of the multiple knowledge management processes proposed in ITIL v3 (Cannon *et al,* 2007), maintaining a well recorded Known Error Database (KEDB) is vital to conduct efficient IT service operations.

The purpose of a Known Error Database (KEDB) is to store the knowledge of tickets– and how they were overcome – to allow quicker diagnosis and resolution when they recur (Cannon *et al*, 2007). The first response to any service outage is to quickly fix the issue and bring the system back up to ensure service availability. The issue would then be sent for root cause analysis, where a decision to implement a change to prevent future occurrences of incidents or update the KEDB with a workaround is taken. The cost benefit analysis determines if there is a business case for a permanent solution.

In our model, KEDB, as an agent, is characterised by the following attributes. Integral to the KEDB is its software efficiency which identifies a new incident and matches with a KEDB record if it exists. We codify the search efficiency of KEDB on a scale of 0 to 1. The number of records/articles in the database is the second attribute. The last attribute is the overall efficiency of KEDB which directly impacts the average resolution time. It is derived from the other two attributes (number of articles and search efficiency).

3.1.4 Governor

A key agent in our model is the Governor, who makes policy decisions at the start of operations on every day. In a real setting, this role is played by the engagement manager. To replicate the cognitive process of decision making by the manager, we adopt Auer's (2003) upper confidence bound algorithm for exploration and exploitation. Originally, designed for modelling the random bandit problem (Robbins, 1952), the algorithm models the problem of a gambler in a room with multiple slot machines and has to decide which slot machine he wants to play in each trail. It is analogous to the problem of engagement manager who has multiple policy options and has to decide which one to adopt each day. As often is the case with humans, while making policy decisions, the algorithm assumes to have only limited knowledge about the rewards associated with each policy choice. Hence, occasionally the algorithm explores various policy options to improve the knowledge about rewards. Exploration, however, does not necessarily improve the current payoff.

To account for the varying levels of exploratory nature of the decision maker, a penalty term that dis-incentivizes is added to the payoff. The penalty term is multiplied by a quantifier that ranges between -1 to 1. The quantifier and penalty term are critical in bringing the exploration and exploitation trade off associated with making policy decisions. The average of payoffs implies the current knowledge of the decision maker and more importantly, facilitates the learning aspect in the algorithm and also guides agent's future exploration. As the agent plays the game more, i.e. gains more experience, his ability to choose the optimal policy increases. Another key aspect of decision making process is the ability to respond to changes in the environment. By using a sliding window that attaches more weightage to newer policy runs, the agent accounts for changes in the environment. A volatile environment mandates a more responsive decision maker; hence a smaller sliding window would be more beneficial. To start with, each policy option is executed once during the initialization period to compute payoffs. The payoff (X) in our model is defined in equation 1.

$$X = (-\lambda * (F * n'(SLA)) + (\lambda - 1) * E) \tag{1}$$

F represents the penalty for each ticket that is not SLA compliant. It is important to note that this penalty is different from the penalty described in previous paragraph. While penalty (F) is to choose policies that minimize tickets out of SLA, the penalty (P) described in the above paragraph is a behavioural parameter of the decision maker. $n(SLA)$ represents the tickets resolved within SLA, $n'(SLA)$ represents the total number of tickets that missed SLA. λ is used to attach relative importance between non compliant SLA tickets and total effort (E) available for resolving tickets. Effort available (E) is the product of number of resources and number of hours each resource can work for and is represented in person-hours. A policy that achieves maximum SLA compliance while consuming minimum effort is ought to have maximum payoff. Expected reward of each policy option is computed as shown in equation 2.

$$Z_{P_n} = X_{avg} + P \tag{2}$$

$$P = B * \ln\left(\sqrt{t/t_i}\right) \tag{3}$$

While X_{avg} is the average payoff of the policy P_n over all the runs in the sliding window, P is the penalty term that introduces the sensitivity to exploratory nature of the agent while making decisions. While B at -1 indicates extreme exploitation, +1 indicates the exploration extreme. Exploitation promotes use of policies that are tried, tested and produced relatively better rewards. Exploration strategy encourages the use of policies that have not been used recently in search higher rewards. B quantifies the exploratory behaviour or risk taking nature of the policy maker.

Table 1: Agents and their Attributes.

Agent	Attributes		Agent	Attributes	
	Static	**Dynamic**		**Static**	**Dynamic**
Resource	Technology tower	Net Effort	Governor	Window Size	Active Policy Set
	Competency	Ticket		B	Payoffs
	Cost	Shift		Λ	
	P(absence)		KEDB	Search Efficiency	Articles
Ticket	Tower	Net Effort			Effectiveness
	KEDB Entry	Resource			
	SLA	Shift			
	Competency				
	Priority				

Table 2: SLA Violations and Average Effort times of Tickets.

	Technology tower 1			Technology tower 2		
Priority	**%**	**Violations**	**Average Effort**	**%**	**Violations**	**Average Effort**
Critical	2.97%	8.46%	28 min	4.59%	9.78%	17 min
High	41.47%	6.37%	146 min	38.97%	8.45%	197 min
Medium	40.60%	5.43%	3346 min	42.64%	6.66%	2876 min
Low	14.96%	3.86%	14547 min	13.80%	4.87%	16543 min

Some of the other key attributes of 'Governor' as an agent are as follows. Window size is to fix the number of periods that are considered for computing average payoff. As discussed earlier, smaller window sizes suit volatile engagements. Active Policy is a dynamic attribute that changes based on the prevailing set of governance policies. λ is used to alter the relative importance attached between effort/cost reduction and better SLA compliance levels.

3.2 Policies

3.2.1 Assignment Policy

Assignment rules define how to assign tickets to resources on the basis of priority, competency and technology tower. Assignment policy decides the order in which incoming tickets would be allocated to a resource and to which particular resource they are assigned to. The allocation of ticket to a resource depends on various factors such as the type of ticket, the expertise level required to resolve the, particular competency required for resolution and whether fungibility across levels and technology towers is present. Fungibility here means resources are free to move across levels and technology towers of support to complete the pending tasks. A fungible structure in production service engagements allows more equitable distribution of work amongst resources leading to higher utilization of resources and lower waiting time for issues to be resolved. However, due to reasons such as geographical distances, shift timings, cost of resources etc. fungibility is not always feasible.

3.2.2 Pre-emption Policy

Pre-emption rules outline the conditions under which a resource can pre-empt the resolution process of the ticket he is currently assigned to pick up another ticket. Pre-emption policy decides whether there would be an interruption to prioritize resolution of some tickets over others in process at any given time. Further, the interruption would be based upon the priority of ticket or SLA time of the ticket or both. Pre-emption policies also decide the way in which overhead caused by pre-emption should be handled.

4 EXPERIMENTAL RESULTS

To feed our simulation model, we used a ticket workload log spanning one month. The total ticket inflow during this period was about 1,839 tickets spread over two supports technology towers of a large financial services provider. The ticket log comprises other relevant information such as arrival times, priority, resolution time, effort time, time spent at each support layer, SLA compliance and reassignment reason. Some basic observations of the ticket log are shown in table 2. In addition, the staffing structure of the engagement is presented in table 3.

To ensure the model conditions are reproduced to the extent possible, the tickets are fed into the model as it is from the ticket log. We avoided deriving distributions from the log and regenerating tickets within the model as the workload remained constant in all the experiments. We evaluate parameters such as SLA compliance, cost of optimized resource set under multiple governor configurations while observing the movement of optimal governance policy set (Table 4).

Table 3: Staffing Structure.

Shift	Technology tower	Levels	Resources
1	1	(1,2,3)	(5,1,3)
	2	(1,2,3)	(3,3,1)
2	1	(1,2,3)	(5,1,3)
	2	(1,2,3)	(3,2,1)
3	1	(1,2,3)	(5,1,3)
	2	(1,2,3)	(2,3,1)
Cost	**$432645**	**SLA**	**95.36%**

Table 4: Ticket Handling Policies.

#	Assignment Policies	#	Pre-emption Policies
A1	No fungibility	M1	No Pre-emption
A2	Fungibility across levels	M2	Pre-emption based on Priority
A3	Fungibility across levels and technology towers	M3	Pre-emption based on SLA expiry

Hybrid Policy Configurations

P1	P2	P3	P4	P5	P6	P7	P8	P9
A1,M1	A1,M2	A1,M3	A2,M1	A2,M2	A2,M3	A3,M1	A3,M2	A3,M3

4.1 Scenario Analysis

The purpose of this exercise is to feed the same ticket log and see the performance under multiple governor configurations. To start with, the three governor configurations mentioned in table 5 are used to see the policy movements. To evaluate these configurations and their impact on SLA compliance and effort reduction, a simulator based on the agent based model discussed in section 3 has been developed in Netlogo. On top of the simulator is an optimizer that was built to produce the optimal resource configuration given a workload, SLA constraints and a set of governor's policy choices. Figures 2a, 2b and 2c show the prevailing policy choice along with the payoff. The impact of changes in the governor configuration is visibly evident in the graphs.

Table 5: Governor Configurations.

	Config 1	Config 2	Config 3
Penalties ($)			
Low	6	10	6
Medium	8	12	7
High	15	14	9
Critical	20	15	11
Governor Parameters			
λ	0.6	0.7	0.1
B	-1.0	0.2	1.0
Window	7 days		

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

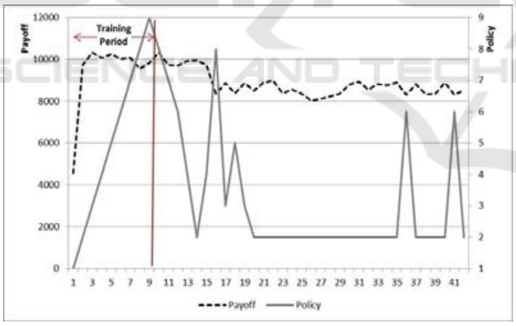

Figure 2a: Policies-Payoffs for Configuration 1.

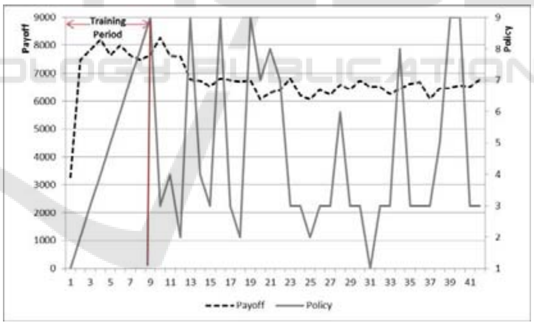

Figure 2b: Policies-Payoffs for Configuration 2.

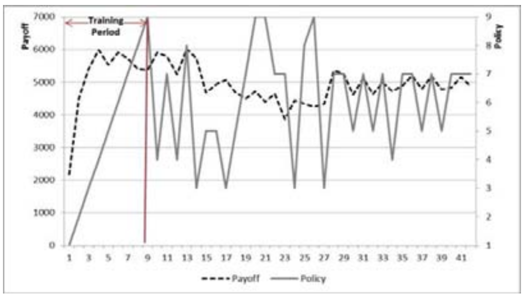

Figure 2c: Policies Payoffs for Configuration 3.

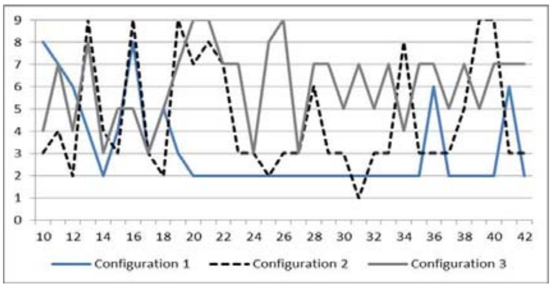

Figure 2d: Policy changes across configurations.

The governor using the terms of reinforcement learning at the start of operations on each day represent the state of the environment and the choice of an alternative represents the action of the learning algorithm. The balance between these two phenomena is shown by the difference in policy choices of configuration 1 and 3, where the value of B varies from one extreme to another. The policy choices (Figure 2d) in each of these configurations may not signify much standalone but when put together with their corresponding SLA compliance levels and cost parameters can provide interesting insights. As shown in table 6, for the same context (ticket workload, priority and SLA norms), changing the Governor configurations can impact the objective realization.

Table 6: SLA, Costs across configurations.

Scenario	SLA Compliance	Cost ($)
Config 1	95.98%	428617
Config 2	94.43%	427343
Config 3	96.87%	441667

λ signifies the relative weights attached to cost and SLA compliance. Therefore, it is expected that Configuration 3 to have more SLA Compliance than Configurations 1 and 2. Similarly, Configuration 2 would focus make policy choices that minimize cost considerations. In contrast, the implications of parameter B, which signifies the exploratory behaviour of the policy maker, are not so straightforward. In the next section, we conduct sensitivity analysis by varying B across the two extremes of exploration and exploitation to understand its impact on SLA compliance and cost objectives.

4.2 Sensitivity Analysis

We divided the spectrum of B from -1 to 1 into a set of 21 values spaced with a difference of .1. To derive the relation between B and Cost, the simulation is run for each of these values of B with different resource configurations (number of resources at each level, technology tower) before zeroing in on the configuration that satisfies SLA constraints with minimum cost. The optimizer that was built to work on the results generated from the simulator outputs the minimum cost.

The second part of the sensitivity analysis is to derive the relation between B and SLA compliance. To conduct this experiment, we have kept the resources constant while varying the parameter B to see the changes in SLA compliance.

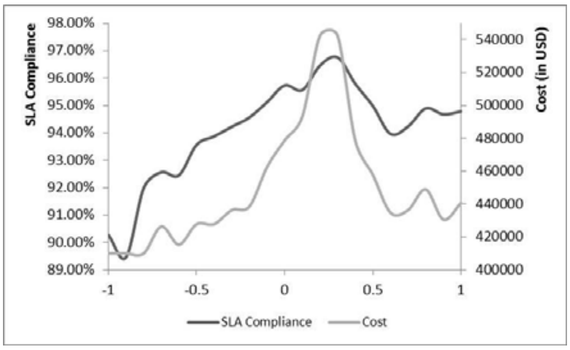

Figure 3: Sensitivity Analysis (B vs Cost and SLA Compliance).

The results are shown in Figure 3. It is interesting to observe the magnitude of changes in both black and grey curves despite the ticket workload remaining constant throughout the sensitivity analyses. Consequently, the impact of governance policy choices on the goal realization is very pronounced. In this case, with the given distribution and frequency of ticket inflow, a value close to .3 yields the best SLA compliance. In comparison, a value of -.7 for B is better suited to minimize the overall resource costs.

5 CONCLUSIONS

We have shown in this study the impact of governance rules in an IT service support engagement on the business level goals such as service availability and cost reduction. The first contribution of this study is to replicate the IT service support system with an agent based model. Central to the model is the Governor agent, which plays the role of manager in an actual setting. Due its parallels with the popular 'Random Bandit' problem, we have borrowed the Upper confidence bound algorithm for exploration and exploitation algorithm to model the cognitive process of the Governor's decision making. The agent is designed to be autonomous and can independently make policy choices based on the environmental variables. The second contribution of this study is to connect the Governor's attributes/behaviour to key business objectives such as SLA compliance and resource costs. Since Governor's attribute configurations have a direct bearing on the policy choices, the link between policy choices and business objectives is incidental. An interesting direction for future research is how to extrapolate the behavioural attributes of the Governor to the manager of an actual IT service support engagement, thereby, establishing an association between the suitability of a manager and the nature of support engagement.

REFERENCES

Assunçao, M. D., Cavalcante, V. F., de C Gatti, M. A., Netto, M. A., Pinhanez, C. S., & de Souza, C. R. (2012, December). Scheduling with preemption for incident management: when interrupting tasks is not such a bad idea. In Proceedings of the *Winter Simulation Conference* (p. 403).

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov), 397-422.

Bartolini, C., & Sallé, M. (2004, November). Business driven prioritization of service incidents. In International *Workshop on Distributed Systems: Operations and Management* (pp. 64-75). Springer Berlin Heidelberg.

Bassamboo, A., Harrison, J. M., & Zeevi, A. (2006). Design and control of a large call center: Asymptotic analysis of an LP-based method. *Operations Research*, 54(3), 419-435.

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

Cannon, D., Wheeldon, D., Taylor, S., & Office of Government Commerce. (2007). *ITIL:[IT service management practices; ITIL v3 core publications][4]. Service operation*. TSO (The Stationery Office).

Casti, J. (1997) *Would-Be Worlds: How Simulation Is Changing the World of Science* (Wiley, New York).

Davidsson, P. (2002). “Agent based social simulation: A computer science view”. *Journal of artificial societies and social simulation*, 5(1).

Gurvich, I., Armony, M., & Mandelbaum, A. (2008). Service-level differentiation in call centers with fully flexible servers. *Management Science*, 54(2), 279-294.

Jennings, N. R. (2000). On agent-based software engineering, *Artificial Intelligence*, 117:277-296.

Jha, A. K., Puvvala, A., Mehta, S., Rai, V. K., & Vin, H. M. (2014). An Agent Based Approach for Effort Estimation in Production Support. *Proceedings of the 25th Australasian Conference on Information Systems, 8th - 10th December, Auckland, New Zealand [179]*.

Lunardi, R. C., Andreis, F. G., da Costa Cordeiro, W. L., Wickboldt, J. A., Dalmazo, B. L., dos Santos, R. L., & Bartolini, C. (2010, April). On strategies for planning the assignment of human resources to it change activities. In 2010 *IEEE Network Operations and Management Symposium*- 2010 (pp. 248-255). IEEE.

Mataric, M. J. (1993). “Designing emergent behaviors: From local interactions to collective intelligence”. In: Proceedings of the *Second International Conference on Simulation of Adaptive Behavior* (pp. 432-441).

Pinedo, M. *Scheduling: theory, algorithms and systems,* Prentice-Hall, Englewood Cliffs, NJ 1995.

Robbins, H. (1985). *Some aspects of the sequential design of experiments. In Herbert Robbins Selected Papers* (pp. 169-177). Springer New York.

Rothkopf, M. H. (1966). Scheduling independent tasks on parallel processors. *Management Science*, *12*(5), 437-447.

Sallé, M. (2004). IT Service Management and IT Governance: review, comparative analysis and their impact on utility computing. Hewlett-Packard Company, 8-17.

Waldrop, M. M. (1992). *Complexity: The Emerging Science at the Edge of Order and Chaos*. New York, NY: Touchstone.

Zaffar, M. A., Kumar, R. L. and Zhao, K. (2008). “Diffusion Dynamics of Open-Source Software in the Presence of Upgrades: An Agent-Based Computational Economics (ACE) Approach”, In: Twenty Ninth International Conference on Information Systems (ICIS 2008) Paris: Paper.55.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS