SYMIAN: a Simulation Tool for the Optimization of the IT Incident Management Process

Claudio Bartolini[1, 2], Cesare Stefanelli[2], and Mauro Tortonesi[2]

[1] HP Research Labs, Palo Alto, CA, USA
claudio.bartolini@hp.com
[2] Engineering Department, University of Ferrara, Ferrara, Italy
{cstefanelli,mtortonesi}@ing.unife.it

Abstract. Incident Management is the process through which IT support organizations manage to restore normal service operation after a service disruption. The complexity of IT support organizations makes it extremely hard to understand the impact of organizational, structural and behavioral components on the performance of the currently adopted incident management strategy and, consequently, which actions could improve it. This paper presents SYMIAN, a decision support tool for the improvement of incident management performance. SYMIAN is a discrete event simulator that permits to test possible corrective measures for the IT support organization before the expensive actual implementation. SYMIAN models the IT support organization as a queuing system, considering both the time spent by operators working on incidents and the time spent when waiting for operator's availability. Experimental results show the SYMIAN effectiveness in the performance analysis and optimization of the incident resolution time for a fictitious organization designed according to real-life experiences.

Keywords: Business-driven IT management (BDIM), decision support, Information Technology Infrastructure Library (ITIL), IT service management, incident management.

1. Introduction

The IT Infrastructure Library (ITIL [1]) is a comprehensive set of concepts and techniques for managing IT infrastructure, development, and operations. Developed by the UK Office of Government Commerce (OGC), ITIL is today the de facto best practice standard for IT service management. Among the processes that ITIL defines, *Incident Management* is the process through which IT support organizations manage to restore normal service operation after a disruption, as quickly as possible and with minimum impact on the business.

Like other IT service operation processes, the incident management process has objectives that are organization-specific and defined by the business management, e.g., compliance with SLAs for some (premium) customers, minimization of economic cost in restoring service, or overall minimization of service disruption interval. The achievement of business objectives in turn requires, at the business

management level, the definition and implementation of strategies in incident management.

IT support organizations need to assess their performance in dealing with service disruptions, in order to verify the effectiveness of their incident management strategies and to evaluate possible alternative strategies. Frameworks such as ITIL and COBIT [2] help by defining objectives for incident management, and usually linking them to simple high-level organization-wide performance metrics such as the mean time to incident resolution. However, the performance analysis of large IT support organizations is non-trivial and might involve a large set of complex and lower-level metrics.

The complexity of IT support organizations and the wide set of metrics to consider make it extremely hard to assess the performance of currently adopted incident management strategies. The evaluation of alternative strategies is even more difficult, as the estimation of potential improvements in incident management requires both an accurate modeling of the IT organization and the identification of critical parameters at the organizational, structural, and behavioral level on which to operate. In particular, the realignment of incident management strategies has to consider a large set of possible operations, such as restaffing (the restructuring of the support organization by increasing or cutting staffing levels, or the transfer of operators around support groups, possibly on retraining), and the implementation of different incident assignment and/or prioritization policies.

The complexity of the incident management domain makes it impossible to treat the performance optimization problem analytically, and calls for simulation-based approaches. In this context, the paper presents **SYMIAN** (SYmulation for Incident ANalysis), a decision support tool based on discrete event simulation. SYMIAN is designed to evaluate and to optimize the performance of the incident management function in IT support organizations.

SYMIAN models the IT support organization as a queuing system, an approach that is particularly well suited for the incident management application domain. In fact, it allows to distinguish the two main components of the time to resolve an incident: *working-time*, and *waiting-time*. Working-time is the time spent by operators working on trouble-tickets (*incidents* in ITIL parlance). Waiting-time is the time spent by trouble-tickets in the queues waiting for technicians to become available to operate over them or to escalate them to other parts of the organization.

SYMIAN allows users to build an accurate model of a real IT support organization and to verify its performance. In addition, SYMIAN permits to play out what-if scenarios, such as adding technicians to a given support group, merging support groups together, experimenting with alternative incident routing and/or prioritization policies, before going through the expensive and time-consuming process of implementing the actual corrective measures.

The SYMIAN tool has been applied for the performance improvement of several case studies representative of the complexity of real-life IT support organizations. The results demonstrated the effectiveness of the SYMIAN-based performance analysis and optimization process.

The paper is structured as follows. Section 2 describes the abstraction of the incident management process and the specification of the associated decision

problem. Section 3 introduces the SYMIAN tool and section 4 sketches both its architecture and implementation. Section 5 presents experimental results obtained by the SYMIAN adoption in the context of a realistic case study. Section 6 reviews related work and compares our approach with it. Finally, Section 7 provides conclusive remarks and future work considerations.

2. Incident Management in IT Support Organizations

A typical IT support organization consists of a network of support groups, each comprising of a set of operators, with their work schedule. Support groups are divided into support levels (usually three to five), with lower level groups dealing with generic issues and higher level groups handling technical and time-consuming tasks. Support groups are further specialized by category of incidents that they deal with (network, server, etc…) and divided into geographies, to ensure prompt incident response (see Figure 1).

In particular, the Help Desk represents the interface for customers reporting an IT service disruption. In response to a customer request, the Help Desk "opens" an incident, sometimes called *trouble-ticket* or simply ticket. The incident is then "assigned" to a specific support group, whose technicians either fully repair the incident or "reassign" it to a different support group (usually escalating to a higher support level). As a result, an incident might have different states and be handled by different support groups throughout its lifetime. At each of these steps, the incident record is updated with the pertinent information, such as current state and related service restoration activity. If, for some reason, customers request the organization to stop working on the incident, the incident is placed in a "suspended" state to avoid incurring into SLO *(Service Level Objective)* penalties. Once the disruption is repaired, the ticket is placed in "closed" state until the end-user confirms that the service has been fully restored. In this case, the incident is "resolved" and its lifecycle ends (see Figure 2).

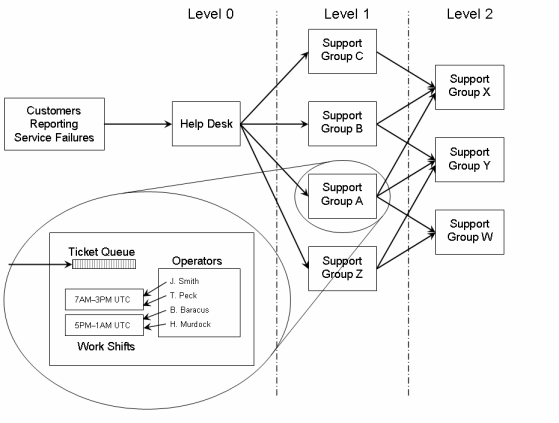

Fig. 1. Conceptual model of the IT support organization for incident management.

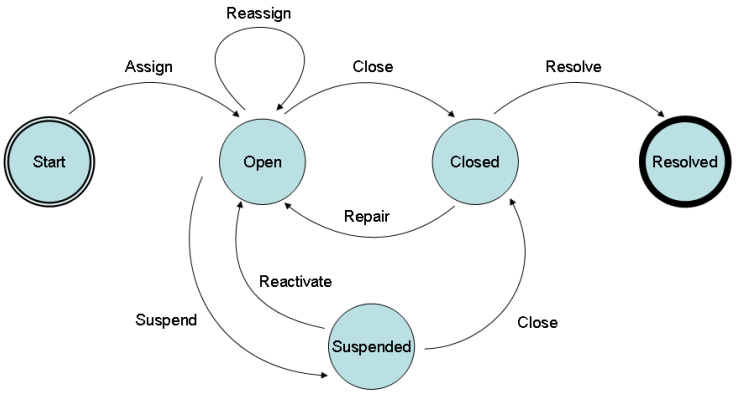

Fig. 2. Incident lifecycle.

The complexity of IT support organizations hinders the verification of the alignment of current organizational, structural, and behavioral processes with the strategic objectives defined at the business management level. In fact, the performance assessment of the incident management function is a very complex procedure which involves the business impact evaluation of the current incident management strategy, through the definition of a set of metrics that allow the objective measurement of performance indicators [3, 4]. Performance analysis and optimization are also organization-specific procedures, since the business impact of service disruptions, and consequently the metrics to consider, vary with the nature of the services and the types of disruptions that occur.

This paper does not consider the processes of business impact analysis and performance metric selection, but instead focuses on the optimization of the organizational, structural and behavioral processes for incident management according to a specified set of metrics. Hence, without loss of generality, it considers for performance optimization the ITIL-recommended objective of service disruption time minimization, and two fundamental and complementary metrics: *Mean Time To (incident) Resolution (MTTR)* and *Mean Incidents Closed Daily (MICD)*.

MTTR and MICD are organization-wide metrics, and as such they provide little insight on the internal dynamics of the organization. A comprehensive performance analysis of the incident management process has to delve into a deeper level of detail. More specifically, it needs to consider both inter- and intra- support groups dynamics, along two orthogonal dimensions: the *effectiveness* of incident routing and the *efficiency* of every single support group in dealing with the incidents. This requires taking into consideration other performance metrics which can evaluate the organization capability to directly forward incidents to the best equipped support groups and the optimality of staff allocation and operator work shift scheduling.

While the application of specific metrics for the performance evaluation of real IT support organization is almost straightforward, it is extremely difficult to evaluate the impact of changes in the organization on these metrics. As a result, the performance assessment of alternative organizations calls for decision support tools enabling *what-if scenario* analysis.

3. The SYMIAN Decision Support Tool

SYMIAN is a decision support tool for the performance analysis and optimization of the incident management function in IT support organizations. In particular, SYMIAN exploits a *discrete event simulator* to reproduce the behavior of IT organizations and to evaluate their incident management performance.

SYMIAN enables its users to play out *what-if scenarios*, allowing them to assess likely improvements in performance without having to go through the expensive and time-consuming process of implementing organizational, structural and behavioral changes. More specifically, SYMIAN allows users to incrementally specify the set of changes to apply to the current organization in order to define an alternative organization configuration that will be tested on a set of performance metrics. For instance, SYMIAN allows modifications such as re-staffing support groups, merging support groups together, experimenting with alternative work shifts, incident routing and/or prioritization policies, or other such actions. SYMIAN guides users all along the optimization process, providing ad hoc visualization of simulation results and, in case a limited set of predefined metrics such as MTTR is considered, explicit tips for the modification of some organization parameters such as the staff allocation.

SYMIAN models the IT support organization (in terms of the number of support groups, the support level, the set of operators, the operator work shifts, the relationships with other support groups, etc.) and permits to define the set of performance metrics to consider for the optimization. SYMIAN then simulates the organization behavior considering a user specified set of incidents, evaluating the desired performance metrics.

At its core, SYMIAN implements an accurate model of the IT support organization. Modeling the incident management function of IT support organizations is an arduous task. In particular, the creation of a realistic model poses two main challenges: the complexity of the IT support organization, and the extremely high volume of incidents and service calls that a typical IT support organization experiences. In addition, the effective adoption of an IT support organization in the context of a decision support tool poses significant constraints on its computational complexity. SYMIAN's model is complex enough to capture the dynamics of a real IT support organization, yet simple enough to allow for an efficient implementation and a user-friendly configuration interface.

SYMIAN models the IT support organization as a *queuing system*. More specifically, the simulated organization behavior emerges from the interaction of its support groups, which are the basic elements of the SYMIAN queuing model. In particular, each support group has a set of operators and a queue of incoming tickets. In turn, every operator has a work shift and is unavailable when off duty. When an operator is idle, he picks the ticket on top of the queue and starts working on it until the operator shift ends or the incident is resolved or cannot be further processed and needs to be forwarded to another support group. In the first case, the operator stops working on the ticket and puts it back in the incoming queue. The ticket will later be extracted from the queue following a configurable prioritization policy. Upon incident closure or escalation, the operator takes another incident from the incoming queue or remains idle if no more incidents exist.

To model the relationships between support groups, and consequently the routing of incidents through the simulated organization, SYMIAN uses a *stochastic transition matrix*. For each support group, the transition matrix describes the probability that incidents of any given category will be forwarded to a specific support group. This model builds on top of the assumption of *memory-less* incident routing, i.e., the probability of incident transition to a specific support group is independent of the history of re-assignments that the incident went through up to that moment. While this assumption allows for a considerable simplification of the model, extensive tests performed with real-life data (using the same dataset as in [5]) on the SYMIAN tool demonstrated that the model behaves with excellent fidelity. A full discussion of the SYMIAN model validation is beyond the scope of the present paper.

Incidents are injected into the system by an incident generation entity which models the aggregate behavior of customer incident reports. An accurate model of the incident arrival/generation process is of critical importance for a realistic simulation. To ensure a realistic input for the simulation, one possibility is to use traces of incidents obtained from the analysis of the operational logs in real IT support organizations. However, considering only real incident traces would limit the applicability of the simulative approach to a small set of predefined input, thus preventing its use to verify how the modeled organization would behave under heavy incident load or under a specific set of incidents following a given inter-arrival or severity pattern. As a result, there is the need to consider synthetic incident generation according to configurable stochastic patterns.

To this end, SYMIAN allows for a highly configurable stochastic incident generation. More specifically, SYMIAN divides incidents in several categories, according to the amount of work they require for service restoration at every support level. In addition, every incident category has several levels of severity, with an increasing (average) time to incident closure or escalation to a higher level support group. Every specific category and severity couple is assigned a random probability distribution which allows the configuration of the amount of work required by incidents. Incident inter-arrival time is also stochastically modeled according to a random variable distribution.

4. SYMIAN: Architecture and Implementation

The architecture of the SYMIAN tool is depicted in Figure 3, that shows its main components: the *User Interface* (UI), the *Configuration Manager* (CM), the *Simulator Core* (SC), the *Data Collector* (DC), and the *Trace Analyzer* (TA).

The User Interface component allows users to load simulation parameters from a file, to change current simulation parameters, to save current simulation parameters to file, and to start simulations. UI provides both an interactive textual and a non-interactive command-line interface.

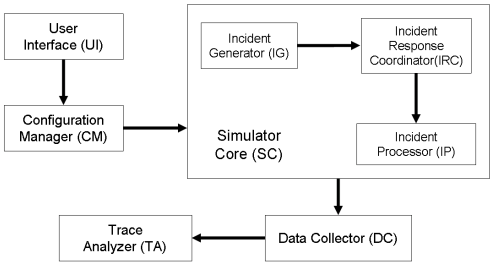

Fig. 3. Architecture of the SYMIAN tool.

The Configuration Manager takes care of the simulator configuration, enforcing the user-specified behaviors, e.g., with regards to verbosity of tracing information, and simulator parameters, e.g., the characterization of incident generation, the number and size of support groups, and the relationships between support groups, in the domain specific model recreated by the Simulator Core component.

The Simulator Core component implements the domain specific model. SC has three sub-components: *Incident Generator* (IG), *Incident Response Coordinator* (IRC) and *Incident Processor* (IP). The Incident Generator generates incidents according to a random distribution pattern which follows user-specified parameters, and injects them into the system. The Incident Response Coordinator receives incidents and dispatches them to the processing domain entities (support groups), which are in turn implemented by the Incident Processor.

The Data Collector component collects data from the simulation that can be post-processed to assess the performance of incident management in the modeled organization. In particular, DC performs an accurate monitoring of support group status, in terms of incoming incident queue size and operator activity, and a careful tracking of incidents status. DC saves its simulation results data in a file that users can then analyze with the Trace Analyzer component.

`SYMIAN` is implemented in the Ruby (*http://www.ruby-lang.org/*) programming language. Ruby was chosen for its remarkable extensibility and its support for meta-programming. The capability to easily redefine the behavior of time-handling classes in the Ruby standard library allowed the implementation of a simulated clock which models the flow of simulation-time in a very similar way to what happens in real life. In addition, Ruby's meta-programming enabled the definition of domain-specific languages and their use in the realization of several simulator components. These have proved to be particularly effective development techniques.

The availability of a wide range of high-quality scientific libraries was also a major reason behind the adoption of Ruby. In particular, `SYMIAN` exploits the GNU Scientific Library (GSL), via the Ruby/GSL bindings, for high-quality random number generation, and it integrates with the Gnuplot data visualization tool to plot some of the simulation results. Finally, `SYMIAN` exploits Ruby facilities to import configuration parameters and export simulation results in the XML, YAML, and CSV formats, in order to ease integration with external software for the automation of multiple simulation runs and with scientific tools for post processing of simulation results.

5. Experimental Results

This section presents an experimental evaluation of the SYMIAN effectiveness in the performance analysis and optimization of the incident management process. More specifically, SYMIAN is applied to minimize the service disruption time in the context of a case study IT support organization, with the constraint of preserving the current number of operators.

As a result, the objectives of the performance improvement process are the maximization of the mean incidents closed daily (*MICD*) metric, as well as the minimization of the mean time to resolution (*MTTR*) metric.

The target of this experimental evaluation is the fictitious incident management organization *INCS'R'US*, which is composed of 3 support levels (0-2), 31 support groups, and 348 operators. The complete characterization of the 31 support groups is presented in Table 1. To limit the complexity of the case study, the routing of incidents in the INCS'R'US organization is assumed to be unidirectional, that is support groups of level N can only receive incidents from support groups of level N-1 and escalate incidents to support groups of level N+1. In addition, an equal probability of incident escalation to each of the support groups of immediately higher level is assumed.

INCS'R'US deals with incidents modeled according to the characterization provided in Table 2. Incidents have 4 categories (A-D) and 3 severity levels (1-3). For every specific combination of incident category and severity, the amount of work that incidents require for service restoration, at every support level, follows a uniform random probability distribution. In Table 2, the abbreviated notation $U(\alpha)$, where $\alpha > 0$, represents the uniform random variable distribution in the $[0, \alpha]$ interval.

Table 1. Support group characterization in the *Incs'R'Us* incident management organization.

Support Level	Support Group (Number of Operators)	Work Shift
0	Help Desk (75)	(25 operators) 7AM-3PM UTC
		(25 operators) 4AM-12PM UTC
		(25 operators) 12PM-8PM UTC
		(10 operators) 5PM-1AM UTC
1	SG1 (15), SG9 (12), SG15 (13), SG18 (5)	7AM-3PM UTC
	SG2 (7), SG10 (7), SG13 (7)	8AM-4PM UTC
	SG3 (15), SG19 (12)	12PM-8PM UTC
	SG4 (4), SG11 (6)	2PM-10PM UTC
	SG5 (14), SG16 (12), SG20 (6)	4AM-12PM UTC
	SG6(12), SG17 (9)	3AM-11AM UTC
	SG7 (5), SG14 (5)	5PM-1AM UTC
	SG8 (6), SG12 (8)	9AM-5PM UTC
2	SG21 (9), SG25 (10)	2PM-10PM UTC
	SG22 (8), SG26 (8)	9AM-5PM UTC
	SG23 (7), SG27 (7)	8AM-4PM UTC
	SG24 (9), SG28 (10)	5PM-1AM UTC
	SG29 (9)	3AM-11AM UTC
	SG30 (6)	4AM-12PM UTC

Table 2. Stochastic characterization of the amount of work time (in seconds) required for incident closure.

	Severity Level 1	Severity Level 2	Severity Level 3
Category A	L0: U(300) L1: 0 L2: 0	L0: U(900) L1: U(240) L2: 0	L0: U(1800) L1: U(900) L2: U(120)
Category B	L0: U(300) L1: U(1200) L2: U(120)	L0: U(600) L1: U(2400) L2: U(240)	L0: U(900) L1: U(3600) L2: U(480)
Category C	L0: U(600) L1: U(150) L2: U(1200)	L0: U(900) L1: U(300) L2: U(2400)	L0: U(1200) L1: U(450) L2: U(3600)
Category D	L0: U(900) L1: U(1200) L2: U(1200)	L0: U(1800) L1: U(4800) L2: U(4800)	L0: U(2400) L1: U(6000) L2: U(6000)

Category A models incidents which mostly require work at support level 0, and a limited amount of work at higher support levels. Category B and C model incidents which require work at every support level, but mostly at support level 1 and 2 respectively. Category D models incidents which require a significant amount of work at every support level. For every incident, category and severity level are randomly chosen, with uniform probability, at generation time. Incident inter-arrival times follow a random exponential probability distribution with an average of 30 seconds.

A first simulation was conducted to evaluate the performance of the current organization. The simulation covered three whole days of simulated time, starting from 2PM UTC. The first 24 hours of simulated time were not considered for the evaluation of the performance metrics, and were introduced only to prime the simulation environment to avoid taking measurements on a cold start. Table 3 (first column) provides the values for the MICD and MTTR performance metrics obtained from the simulation. The table also shows the Mean Work Time (MWT) metric, defined as the mean work time per closed incident, as an indication on the amount of work spent on service restoration.

By analyzing the variation of the incident queue size at every support group using both SYMIAN graphical visualization and time series analysis functions, it was easy to realize that support groups SG1, SG4, SG7, SG8 and SG14 at support level 1 and support group SG30 at support level 2 were a major performance bottleneck, while the Help Desk and support groups SG3 and SG17 were oversized. As an example of the effectiveness of visual analysis to locate performance bottlenecks, Figure 4 (a) plots the variation of incident queue size at support group SG30.

Table 3. Performance metrics from the first and second simulation.

	First simulation	Second simulation
Total incidents generated	8609	8609
Incidents generated after warm-up	5728	5728
MICD	1811	2002
MTTR (in seconds)	53423	47047
MWT (in seconds)	L0: 508, L1: 809, L2: 784	L0: 506, L1: 811, L2: 773

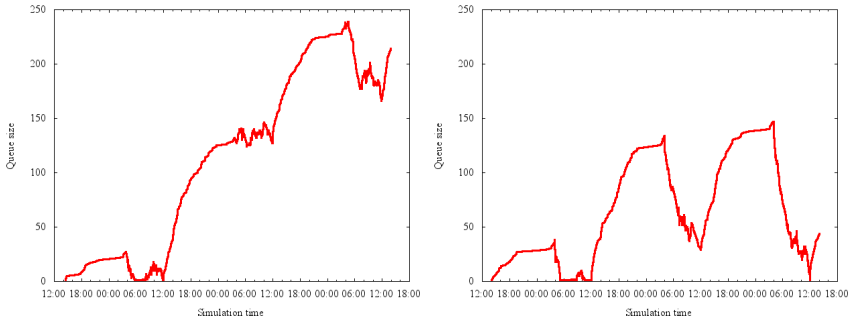

Fig. 4. Incident queue size at support group SG30 during the first (a) and second (b) simulation.

To improve the organization performance, 8 operators were transferred from the Help Desk to support groups SG1, SG4, SG7, and SG8 (2 operators for each group), 3 operators were transferred from support group SG3 to support group SG14, and 2 operators were transferred from support group SG17 to support group SG30. A new simulation was then launched to assess the performance of the new organization. Table 3 (second column) and Figure 4 (b) provide respectively the performance metrics and the variation of incident queue size at support group SG30 for the new simulation.

The results of the second simulation proved that the reallocation of operators was very effective in improving the whole system performance. In particular, the INCS'R'US organization exhibited a 10.5% improvement of the MICD and a 11.9% decrease of the MTTR.

Although the target of the previous performance optimization experiment is a fictitious organization, the case study was carefully designed to be representative of the complexity of real-life IT organizations. Therefore, the simulation results demonstrate the effectiveness of the SYMIAN tool for the performance optimization of the incident management function in IT support organizations.

6. Related Work

The present work contributes to the up and coming research domain of Business-driven IT management (BDIM), which builds on the tradition of the research in network, system and service management. BDIM has been defined as "*the application of a set of models, practices, techniques and tools to map and to quantitatively evaluate interdependencies between business performance and IT solutions – and using the quantified evaluation – to improve the IT solutions' quality of service and related business results*". For a thorough review of BDIM, see [6].

Some notable early works in BDIM include applications to change management [7, 8, 9], capacity management [10, 11, 12], network security [13], and network configuration management [14]. All these research efforts (possibly with the exception of [14]), limit their scope to the *technology* dimension of IT management,

thereby focusing on the fine tuning of systems configuration and on the introduction of automation as means to improve the IT management processes.

The present work, instead, belongs to a recently emerged research area that focuses on the other two fundamental dimensions of IT management: *people* and *processes*. The interest on this topic arose as researchers started analyzing the relationships between people, processes and technological optimization and the impact of automation and process complexity on labor cost. As a representative example, we cite Diao et al.'s recent research effort addressing the very important question of when does it make sense to automate processes based on metrics of process complexity [15, 16]. The main difference between our approach and theirs is that our focus is in achieving significant improvements in the performance of the organization through decision support and simulation techniques. In this context, in previous works we have extensively studied the business impact of incident management strategies [3, 4], using a methodology that moved from the definition of business-level objectives such as those commonly used in balanced scorecards [17]. With respect to those works, this paper follows a novel approach that the first time proposes and implements detailed modeling of the inner functioning of the IT support organization to support what-if scenario analyses.

The analysis of the incident management process and the IT support organization model that we present in this paper share is founded on our work presented in [5]. However, here we push our modeling effort far beyond the definition of metrics for the performance assessment of IT support organizations that we conducted in [5], all the way to the design and implementation of the SYMIAN decision support tool.

7. Conclusions and Future Work

The performance optimization of large-scale IT support organization can be extremely complex and might require additional help from decision support tools. This paper presented the SYMIAN tool for the performance optimization of incident management in IT support organizations. The application of SYMIAN in case studies expressively designed to capture the complexity of real-life IT support organizations demonstrated the tool effectiveness in the difficult performance analysis and improvement process.

Future versions of SYMIAN will be complemented with the application of automated techniques for the optimization of parameters, e.g., staff allocation, in the context of specified performance metrics. The IT support organization model implemented in SYMIAN is also currently being extended to consider operators with *skills* that skew their expected working time for incidents of a given category and priority policies in extracting incidents from queues.

Finally, a more comprehensive version of the SYMIAN tool will link performance optimization metrics with key performance indicators or impact metrics that are meaningful at the business level.

References

[1] IT Infrastructure Library, "ITIL Service Delivery" and "ITIL Service Support", OGC, UK, 2003.

[2] IT Governance Institute, "*COBIT 3rd Edition*", 2000, www.isaca.org/COBIT.htm

[3] Bartolini, C., and Sallé, M., "Business Driven Prioritization of Service Incidents", In *Proceedings of Distributed Systems Operations and Management* (DSOM) 2004.

[4] Bartolini, C., Sallé, M., Trastour, D., "IT Service Management driven by Business Objectives – An Application to Incident Management", *In Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, Apr. 2006

[5] G. Barash, C. Bartolini, Liya Wu, "Measuring and Improving the Performance of an IT Support Organization in Managing Service Incidents", in proc. 2nd IEEE Workshop on Business-driven IT Management (BDIM 2007), Munich, Germany, 2007.

[6] Moura, A., Sauvé, J., Bartolini, C., "Research Challenges of Business–Driven IT Management", *In Proceedings of the 2nd IEEE / IFIP International Workshop On Business-Driven IT Management (BDIM 2007)*, Munich, Germany

[7] Keller, A., Hellerstein, J., Wolf, J.L., Wu, K. and Krishnan, V., "The CHAMPS System: Change Management with Planning and Scheduling", In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium* (NOMS 2004), IEEE Press, April 2004

[8] Sauvé, J., Rebouças, R., Moura, A., Bartolini, C., Boulmakoul, A., Trastour, D., "Business-driven decision support for change management: planning and scheduling of changes", *In Proceedings of the DSOM 2006*, Dublin, Ireland

[9] Trastour, D., Rahmouni, M., Bartolini, C., "Activity-Based Scheduling of IT Changes" *In Proceedings of First International Conference on Autonomous Infrastructure, Management and Security*, *AIMS 2007*, Oslo, Norway, 73-84

[10] Aiber, S., Gilat, D., Landau, A., Razinkov, N., Sela, A. and Wasserkrug, S. "Autonomic Self–Optimization According to Business Objectives"; *In Proceedings of the International Conference on Autonomic Computing*, 2004.

[11] Menascé, D., Almeida, V.A.F., Fonseca, R. and Mendes, M.A., "Business-Oriented Resource Management Policies for e-Commerce Servers", *Performance Evaluation 42*, Elsevier Science, 2000, pp. 223-239.

[12] Sauvé, J., Marques, F., Moura, A., Sampaio, M., Jornada, J. and Radziuk, E., "SLA Design from a Business Perspective", In *Proceedings of DSOM* 2005.

[13] Wei, H., Frinke, D., Carter, O., et al. "Cost–Benefit Analysis for Network Intrusion Detection Systems", In *Proceedings of the 28th Annual Computer Security Conference*, October 2001.

[14] Boutaba, R., Xiao, J. and Aib, I., "CyberPlanner: A Comprehensive Toolkit for Network Service Providers", *in Proceedings of the 11th IEEE/IFIP Network Operation and Management Symposium (NOMS 2008)*, Salvador de Bahia, Brazil.

[15] Diao, Y., Keller, A., Parekh, S., Marinov, V. "Predicting Labor Cost through IT Management Complexity Metrics" *in Proceedings of the 10th IEEE/IFIP Symposium on Integrated Management (IM 2007)*, Munich. Germany.

[16] Diao, Y., Bhattacharya, K., "Estimating Business Value of IT Services through Process Complexity Analysis", *in Proceedings of the 11th IEEE/IFIP Network Operation and Management Symposium (NOMS 2008)*, Salvador de Bahia, Brazil.

[17] Kaplan, R., and Norton, D., "The Balanced Scorecard: Measures that Drive Performance", Harvard Business Review, 70(1), 1992, pp.71-79.